



Integració d'un sistema documental de xarxes de fibra òptica a la xarxa guifi.net (fiberfy)

Treball de Fi de Grau - Especialitat de TI

Roger Garcia i Ferré

Director: Lluís Dalmau Junyent

Ponent: Llorenç Cerdà Alabern



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



Facultat d'Informàtica de Barcelona

Agraïments

En primer lloc m'agradaria agrair als meus pares Neus i Sergi la seva insistència perquè continués els estudis d'enginyeria en aquells moments d'incertesa vital.

Voldria agrair a la meva parella Cinta la seva infinita paciència per escoltar les meves constants històries sobre el projecte i donar-me el suport per seguir.

Voldria agrair la feina feta pel meu predecessor en el projecte fiberfy: Agustí Moll, que m'ha ajudat a entendre el prototip i m'ha donat un cop de mà quan ho he necessitat.

M'agradaria agrair a la gent de Guifi.net de la UPC: Roger Pueyo i Roger Baig, la seva confiança en mi per ocupar el càrrec de desenvolupador a la fundació.

També m'agradaria agrair al professor Llorenç Cerdà per haver-me parlat del projecte Guifi.net i ajudat a instal·lar el meu primer node de Guifi.net. Sense ell no hauria après en tanta profunditat com funcione les xarxes de computadors.

Vull donar les gràcies a tots els treballadors de la Fundació Guifi.net que també han estat una peça clau per a la realització d'aquest projecte.

M'agradaria agrair a l'associació eXO i en especial: Víctor Oncins i Pedro Vílchez per haver-me cedit un grapat de direccions IPv4 públiques per a la realització d'experiments.

Finalment m'agradaria agrair a l'Antoni Puigdomènech per cedir-me un racó de casa seva per a la instal·lació del servidor de testing.

Resum

Aquest document és la memòria final del Treball de fi de Grau (TFG) de l'especialitat de Tecnologies de la Informació del grau en Enginyeria Informàtica per la Facultat d'Informàtica de Barcelona. El treball s'ha fet íntegrament a la Fundació Guifi.net. Guifi.net va començar sent una xarxa de telecomunicacions comunitària basada en enllaços de ràdio sense fils. A partir de l'any 2010 quan el projecte ja tenia una mida considerable es va decidir apostar també per aplicar el model de Guifi.net a les comunicacions òptiques i així plantejar una alternativa a les operadores tradicionals arribant a moltes zones rurals que en aquell moment tenien connexions de molt baixa qualitat. En aquests 10 anys, des del primer desplegament de fibra òptica, s'ha portat fibra a un bon nombre de zones de la geografia catalana convertint Guifi.net en una alternativa al nivell de les grans operadores de telecomunicacions. El projecte exposat en aquesta memòria pretén donar resposta a les demandes dels participants del projecte a l'hora d'inventariar aquesta xarxa de fibra òptica de nova creació. Es tracta des del desenvolupament d'una solució a la seva posada en marxa i integració amb els sistemes existents.

Resumen

Este documento es la memoria final del Trabajo de Fin de Grado (TFG) de la especialidad de Tecnologías de la Información del Grado en Ingeniería Informática por la Facultad de Informática de Barcelona. El trabajo se ha realizado íntegramente en la Fundación Guifi.net. Guifi.net empezó siendo una red de telecomunicaciones comunitaria basada mayoritariamente en enlaces inalámbricos. A partir del año 2010 cuando el proyecto ya tenía un tamaño considerable se decidió apostar también por aplicar el modelo de Guifi.net a las comunicaciones ópticas y así plantear una alternativa a las operadoras tradicionales, llegando a muchas zonas rurales que en aquel momento tenían conexiones de muy baja calidad. En estos 10 años, desde el primer despliegue de fibra óptica, se han construido nuevas redes en un gran número de zonas en Cataluña convirtiendo Guifi.net en una alternativa al nivel de los grandes operadores de telecomunicaciones. El proyecto expuesto en esta memoria pretende dar respuesta a la demanda de los participantes del proyecto a la hora de hacer inventario de esta nueva red óptica. Aquí se trata desde el desarrollo de las soluciones software, su puesta a punto a su integración en los sistemas actuales.

Abstract

This document covers the Master Thesis for the Bachelor's degree in Informatics Engineering, Major in Information Technologies by Facultat d'Informàtica de Barcelona (Barcelona School of Informatics) done in Guifi.net foundation. Guifi.net started as a community network in 2004 based initially in wireless links using Wi-Fi technologies. Since 2010, as the project was widely adopted in Catalonia, the participants decided to start building optical fiber infrastructure using the same community model. This action converted Guifi.net project to an alternative in terms of quality to the other conventional ISP, specially in rural areas where the conventional companies don't invest. The project contained in this report shows the software as a solution to respond the participant's demands in terms of infrastructure inventory. This covers from the application prototype to the software in production. It also includes an analysis for every platform used to integrate all the system.

Índex

| | | |
|----------|---|-----------|
| 1 | Context | 1 |
| 1.1 | Introducció | 1 |
| 1.1.1 | Que és Guifi.net? | 1 |
| 1.1.2 | El projecte i una mica d'història | 1 |
| 1.2 | La xarxa | 3 |
| 1.2.1 | La fibra òptica | 3 |
| 1.2.2 | La documentació de la xarxa | 4 |
| 1.3 | Estat de l'art | 4 |
| 1.3.1 | Possibles solucions | 4 |
| 1.3.2 | Conclusions | 5 |
| 2 | Formulació del problema | 7 |
| 2.1 | Problema | 7 |
| 2.2 | Objectiu | 7 |
| 3 | Abast del treball | 9 |
| 3.1 | Realització | 9 |
| 3.2 | Possibles imprevistos | 10 |
| 4 | Metodologia i seguiment | 11 |
| 4.1 | Mètode de treball | 11 |
| 4.2 | Mètodes de validació | 12 |
| 5 | Recursos | 13 |
| 5.1 | Recursos personals | 13 |
| 5.2 | Recursos materials | 13 |
| 5.3 | Recursos <i>software</i> | 13 |
| 6 | Planificació temporal | 15 |
| 6.1 | Durada del projecte | 15 |
| 6.2 | Tasques | 15 |
| 6.3 | Dependències entre tasques | 16 |

| | | |
|-----------|---|-----------|
| 7 | Valoració d'alternatives i pla d'acció | 19 |
| 7.1 | Riscs | 19 |
| 7.2 | Pla d'acció | 19 |
| 8 | Implementació | 21 |
| 8.1 | Entorns de desenvolupament | 21 |
| 8.1.1 | La tecnologia Docker | 21 |
| 8.1.2 | Els entorns dins de Guifi.net | 22 |
| 8.2 | Backend (fiberfy-server) | 25 |
| 8.2.1 | Contextualització | 25 |
| 8.2.2 | Canvi de framework | 26 |
| 8.2.3 | Canvi de base de dades | 26 |
| 8.2.4 | Integració amb LDAP | 28 |
| 8.2.5 | Implementació de l'importador d'infraestructura | 31 |
| 8.2.6 | API REST | 32 |
| 8.3 | Frontend (fiberfy-nt) | 34 |
| 8.3.1 | Contextualització | 34 |
| 8.3.2 | Nova arquitectura | 34 |
| 8.3.3 | Llibreries utilitzades | 36 |
| 8.4 | Servei de mapes (guifimaps) | 37 |
| 8.4.1 | Contextualització | 37 |
| 8.4.2 | Millores aplicades | 37 |
| 9 | Posada en marxa | 39 |
| 9.1 | Entorns físics | 39 |
| 9.1.1 | Testing | 39 |
| 9.1.2 | Producció | 39 |
| 9.2 | Plataforma de virtualització | 39 |
| 9.2.1 | Introducció a la virtualització | 40 |
| 9.2.2 | Anàlisi de plataformes | 40 |
| 9.3 | Entorns virtuals | 41 |
| 9.3.1 | Creació de la instància (testing) | 41 |
| 9.3.2 | Sistema operatiu | 44 |
| 9.3.3 | Software instal·lat i configuracions | 44 |
| 9.4 | Monitoratge | 45 |
| 9.4.1 | Instal·lació i configuració | 46 |
| 9.5 | Networking (testing) | 48 |
| 9.5.1 | Carrier service | 48 |
| 9.5.2 | Xarxa Local | 49 |
| 10 | Gestió econòmica | 51 |
| 10.1 | Recursos humans | 51 |
| 10.2 | Recursos <i>hardware</i> | 52 |

| | | |
|-----------|---------------------------------|------------|
| 10.3 | Recursos de <i>software</i> | 52 |
| 10.4 | Altres | 52 |
| 10.5 | Resum de costos | 53 |
| 10.6 | Control de gestió | 53 |
| 11 | Sostenibilitat | 55 |
| 11.1 | Ambiental | 55 |
| 11.2 | Econòmica | 55 |
| 11.3 | Social | 56 |
| 12 | Conclusions | 57 |
| 12.1 | Resultats | 57 |
| 12.2 | Millores de futur | 57 |
| | Apèndixs | 59 |
| A | Manual fiberfy-nt | 59 |
| B | Gantt | 79 |
| C | Model de dades | 80 |
| D | Docker images | 82 |
| D.1 | docker-fiberfy | 82 |
| E | Docker compositions | 85 |
| E.1 | drupal6+guifimaps+fiberfy-sails | 85 |
| F | fiberfy-server | 87 |
| F.1 | Models | 87 |
| F.2 | Serveis | 109 |
| F.3 | API | 110 |
| F.4 | Importador | 132 |
| G | Posada a punt | 136 |
| G.1 | Systemd unit | 136 |
| | Llista d'acrònims | 137 |
| | Bibliografia | 137 |

Índex de figures

| | | |
|------|--|----|
| 1.1 | El mapa actual de Guifi.net | 2 |
| 1.2 | Una mànega amb fibres òptiques | 3 |
| 4.1 | El mètode scrum gràficament. | 11 |
| 4.2 | Una pizarra <i>kanban</i> | 12 |
| 8.1 | Arquitectura de la tecnologia Docker | 22 |
| 8.2 | Tipus de punts de muntatge en Docker | 23 |
| 8.3 | Arquitectura de la composició pel <i>fiberfy</i> | 25 |
| 8.4 | Arquitectura MVC | 27 |
| 8.5 | Estructura de directoris de <i>fiberfy-server</i> | 28 |
| 8.6 | Canvi de l'entitat <i>Site</i> | 29 |
| 8.7 | Integració de l'entitat <i>Zone</i> | 30 |
| 8.8 | Exemple de directori LDAP | 31 |
| 8.9 | Swagger UI mostrant API <i>fiberfy-server</i> | 33 |
| 8.10 | Logotips de llibreries <i>frontend</i> | 35 |
| 8.11 | Esquema de funcionament Vuex | 36 |
| 8.12 | Esquema de funcionament Guifimaps amb les millores | 37 |
| 9.1 | Pila de sistemes en virtualització complerta | 40 |
| 9.2 | Dashboard de la plataforma Proxmox VE | 42 |
| 9.3 | Formulari de creació de màquines virtuals (1) | 43 |
| 9.4 | Formulari de creació de màquines virtuals (2) | 43 |
| 9.5 | Logotip de Debian | 44 |
| 9.6 | Logotip de monit | 46 |
| 9.7 | Captura inicial de monit | 47 |
| 9.8 | Diagrama sobre la topologia de xarxa (L2) | 49 |
| 9.9 | Taula amb totes les interfícies del host <i>puigdemont</i> | 49 |
| 12.1 | <i>Gantt</i> del projecte | 79 |
| 12.2 | Model de dades <i>fiberfy</i> | 81 |

Capítol 1

Context

1.1 Introducció

Les telecomunicacions s’han convertit en pocs anys en un servei de primera necessitat per a les persones, com ho va ser al segle XX l’electricitat i el telèfon. En moltes zones rurals l’accés a aquestes tecnologies no ha estat disponible fins que un grup de persones s’ha posat d’acord per tal de promoure ells mateixos la construcció de xarxes. D’aquesta manera es va crear el projecte **Guifi.net** l’any 2004 a Gurb, Osona (avui fa quasi 15 anys).

1.1.1 Que és Guifi.net?

Guifi.net és un projecte tecnològic, social i econòmic impulsat des de la ciutadania que té per objectiu la creació d’una xarxa de telecomunicacions oberta, lliure i neutral basada en un model de comuns. El desenvolupament d’aquesta infraestructura mancomunada facilita l’accés a les telecomunicacions en general i a la connexió a Internet de banda ampla en particular, de qualitat, a un preu just i per a tothom. A més, genera un model d’activitat econòmica col·laborativa, sostenible i de proximitat.[15]

A la figura 1.1 podem veure el mapa actual de la xarxa.

1.1.2 El projecte i una mica d’història

Inicialment aquesta xarxa de comuns es va crear en la seva totalitat utilitzant enllaços inal·lambrics sobre tecnologies obertes (**IEEE 802.11** [9]) més coneguda com a **Wi-Fi**. D’aquesta manera s’encoratjava a la gent que vivia en zones on no hi havia un bon accés a Internet que es muntés les seves pròpies antenes i s’unís a la xarxa de Guifi que s’estava creant. D’aquesta manera es creaven proxies d’Internet en els punts on si que era possible tenir línies **DSL** o de característiques similars i es compartien amb els usuaris de la xarxa que estaven en zones aïllades i no podien disposar d’aquest tipus de serveis.

Per tal de promoure el projecte més enllà de la zona on s’havia formulat (Gurb, plana de Vic) els promotors de la iniciativa van crear un portal (<https://guifi.net>) on els usuaris podien

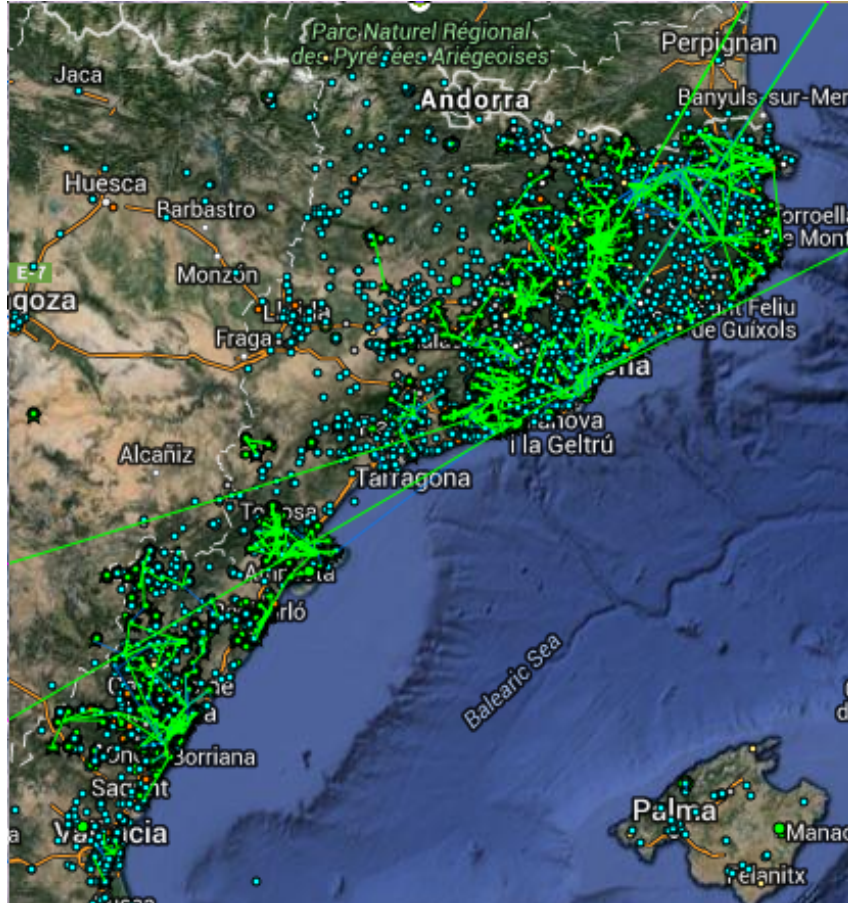


Figura 1.1: El mapa actual de Guifi.net

documentar els seus nodes (llocs on hi havia antenes) i organitzar la xarxa (Direccionament IP, protocols d'enrutament, manuals per a nous usuaris, xat, llistes de correu...)

Gràcies a la col·laboració d'una comunitat creixent al voltant de la idea de fer una xarxa comunitària per a tothom aquest projecte es va estendre ràpidament per a moltes zones de Catalunya que patien el mateix problema que a Osona i també a zones (com Barcelona) on no hi havia un problema greu de connectivitat però si un interès de molta gent en apostar per a una xarxa sobirana i compartida.

L'any 2007 la Generalitat de Catalunya va reconèixer la labor de la comunitat de Guifi.net tot otorgant-li el premi nacional de telecomunicacions [24] [18]. En aquell moment la comunitat va veure la necessitat de dotar d'una estructura jurídica al projecte i es va decidir crear la fundació per a la xarxa oberta, lliure i neutral Guifi.net.

Des d'aquell moment la xarxa ha anat creixent de forma exponencial i a partir de l'any 2009 es comença a desplegar fibra òptica per a donar servei a usuaris finals, empreses, administracions... En aquell moment la fundació es constitueix com a **ISP** majorista, membre del **RIPE** i entra al punt neutre de Catalunya (**CATNIX**).

Un altre canvi important des d'aleshores va ser la creació de la figura dels operadors dins de

Guifi.net. Organitzacions que desenvolupen una activitat econòmica dins del projecte Guifi.net i que presten serveis a usuaris i empreses. Degut a la professionalització del projecte la fundació Guifi.net actua com un regulador del projecte (Governança), un àrbitre que ajuda en la solució de conflictes i també alhora té una funció prestadora de serveis en aquestes organitzacions.

1.2 La xarxa

Per tal d'accedir a Internet les xarxes de telecomunicacions es poden segmentar en 3 grans grups:

- Xarxes d'accés o última milla: És la connexió final dels usuaris. Representa el 90% del cost de la xarxa.
- Transport territorial: Recull el trànsit d'un territori (Per exemple comarca) i el porta fins al punt on l'operador té la seva infraestructura.
- Trànsit global: És tot el trànsit que s'envia als punts d'intercanvi d'Internet (en el nostre cas al **CATNIX**) i també el trànsit que s'enruta cap als *carriers* internacionals.

En el cas de Guifi.net ens hem centrat en l'accés als usuaris o última milla, en el cas del transport territorial s'intenta utilitzar infraestructures de tercers ja construïdes per evitar duplicat d'infraestructura.

1.2.1 La fibra òptica

Inicialment l'accés a la última milla es feia utilitzant radioenllaços però des de l'any 2009 s'ha començat a desplegar xarxa de fibra òptica per als usuaris.

La fibra òptica és un filament flexible de secció circular fet d'un tipus de vidre o plàstic capaç de transportar feixos de llum en el seu interior.[10]

A la figura 1.2 podem veure una mànega amb fibres a dins.

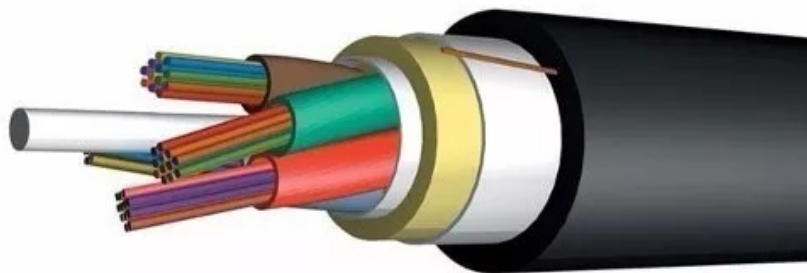


Figura 1.2: Una mànega amb fibres òptiques

Aquest tipus de desplegaments de fibra òptica es coneixen com a **FTTH**. Les sigles **FTTH** signifiquen Fiber-to-the-Home això vol dir que cada usuari disposa d'una línia de fibra òptica

que li arriba fins a casa seva. En el cas de Guifi.net majoritàriament les xarxes de fibra òptica s'implementen dins del protocol **GPON** que ens permet dividir de forma passiva la fibra utilitzant *splitters*. [13]

1.2.2 La documentació de la xarxa

L'èxit del projecte Guifi.net va venir donat pel portal que oferia, on els seus usuaris podien especificar com es col·locaven les antenes i les configuracions de xarxa que tenien. És molt important per tant poder documentar correctament les xarxes de les que es disposa. A dia d'avui encara no es disposa d'una eina satisfactòria per a realitzar de forma eficient la documentació de les xarxes de fibra òptica.

1.3 Estat de l'art

Per tal de documentar fibra òptica es necessiten unes eines que ens permetin treballar amb cartografia i a l'hora definir esquemes de xarxa. Al mercat hi ha un bon grapat de productes que compleixen amb aquestes premisses. La majoria de productes per cobrir aquesta demanda els desenvolupen empreses americanes i estan pensats per petits i mitjans **ISP**. Les grans teleoperadores que dominen el mercat probablement tenen software propi per cobrir aquestes necessitats.

1.3.1 Possibles solucions

VETROFiberMap™

Es tracta d'un producte desenvolupat a estats units que pretén ser una solució a la documentació de xarxes de fibra òptica per a operadors petits, mitjans i xarxes comunitàries (com és el cas de Guifi.net). Les característiques més rellevants del producte en el context d'aquest treball són:

- Es tracta d'un sistema basat en **SaaS**
- És multiplataforma i compatible amb dispositius mòbils.
- Està basat en una **REST Api**
- És compatible amb els estàndards **GIS** més coneguts, inclòs **GeoJSON**
- Té un sistema de preus basat en l'ús i flexible.

A la cita [5] es pot trobar la web del fabricant del software.

FiberBase

En aquest cas el producte no sembla tenir un marketing tant ben plantejat com l'anterior. No fa referències a quin tipus d'organitzacions va dirigit el producte sinó que especifica a quins rols dins de les organitzacions va dirigit el sistema.

- Permet Copy&Paste
- Basat en Oracle
- Disponible al Cloud
- No s'especifica una política de preus (Es demana contactar amb ells).

A la cita [4] es pot trobar la web del fabricant del software.

1.3.2 Conclusions

Després d'analitzar les diferents alternatives n'he extret les següents conclusions:

- No hi ha cap software disponible amb les característiques desitjades, mencionades anteriorment, al nostre país.
- Totes les alternatives vistes estan basades en sistemes propietaris i cap d'elles disposa d'una llicència lliure. Això tenint en compte que la fundació Guifi.net és una entitat sense ànim de lucre i compromesa amb el software lliure és un handicap.
- Els preus d'aquests productes són força elevats.
- En alguns casos les aplicacions ofereixen més funcionalitats de les requerides.
- No hi ha cap sistema que s'adapti al model econòmic que impera dins del projecte Guifi.net.

Després d'això la decisió presa ha estat la de dissenyar el nostre propi sistema documental.

Capítol 2

Formulació del problema

2.1 Problema

Des de la fundació Guifi.net necessitem un sistema que ens permeti documentar els desplegaments de fibra òptica que realitzen els operadors que es troben dins del projecte. Aquests, juntament amb el patronat de la fundació, actuen com a *stakeholders* definint els requeriments del projecte. Per tant el producte va dirigit clarament a aquests operadors. Com que no hi ha cap alternativa oberta i que pugui encaixar en el model multioperador implementat i sobretot amb un preu contingut es decideix desenvolupar la nostra pròpia solució.

2.2 Objectiu

El nostre objectiu és crear una plataforma que ens permeti documentar correctament les xarxes de fibra òptica tant a nivell físic (obra cívica) com la infraestructura de xarxa implicada a nivell 1 del model **OSI**.

Aquí s'enumeren els principals requeriments del sistema:

- Ha de ser *Open Source*. Per principis fundacionals, tot el software que és desenvolupat dins de la fundació s'ha de lliurar amb alguna llicència d'aquest tipus (GPL, AGPL, MIT...)
- Ha d'estar basat en una aplicació web
- Ha de tenir una **API Rest**. Això és important per tal de comunicar el servei amb altres serveis.
- Ha de permetre integració amb la base de dades actual de Guifi.net i amb l'operativa econòmica de Guifi.net (model de compensacions)
- S'ha d'aprofitar el prototip en la mesura del possible per reduir costos d'implementació.

Capítol 3

Abast del treball

3.1 Realització

Cal remarcar que a l'inici d'aquest treball el projecte ja havia estat començat per una altra persona i per tant s'haurien d'afegir uns quants requeriments donats. Partim de l'existència d'un prototip de l'aplicació que volem, amb unes limitacions de disseny importants. No obstant la idea de l'aplicació ha estat validada pels diferents actors. Aleshores els objectius del treball seran els següents:

1. S'han de facilitar els entorns de desenvolupament per tal de que es puguin incorporar en un futur nous programadors. Aquesta tasca es farà utilitzant **Docker**[8]
2. S'ha de canviar la plataforma del servidor per una suportada. Actualment la plataforma triada no satisfà els criteris de fiabilitat requerits.
3. S'ha de refer tot el *frontend* del sistema. L'actual no permet escalar correctament ni afegir noves funcionalitats de forma eficient.
4. Integració amb el model de dades de Guifi.net. Actualment la base de dades és autosuficient i no és compatible amb el model de Guifi.net.
5. Ha de permetre importar dades en estàndards de **GIS** com per exemple **KMZ** (per facilitar la migració de mapes)
6. Ha de permetre exportar les dades amb formats **GIS** estàndards i integrar amb sistemes **WMS**
7. S'ha de fer Q&A (testos unitaris, integració contínua)
8. S'ha de posar el sistema en producció al datacenter de Guifi.net

Tant el *backend* com el *frontend* estan escrits usant **JavaScript**. Això es mantindrà d'aquesta manera ja que al utilitzar un mateix llenguatge ens permet ser més eficients i eficaços en el projecte.

3.2 Possibles imprevistos

Al tractar-se d'un projecte que té aplicacions en el món professional hi poden sorgir un seguit d'imprevistos que en dificultin la seva posada en marxa:

- Problemes d'integració. Al ser un sistema que ja està en marxa un dels problemes amb els que ens podem trobar és la dificultat d'integrar els components actuals del sistema (base de dades, mapes...) a la nova aplicació.
- Manca de personal. És un projecte força ambiciós i la seva dotació en personal és baixa per l'objectiu que es vol assolir. Aquest punt pot alterar-se en un futur pròxim si es decideix contractar més personal.

Capítol 4

Metodologia i seguiment

4.1 Mètode de treball

A la realització d'aquest projecte i tenint en compte que ens interessa anar proveint funcionalitats de forma progressiva als nostres *stakeholders*, utilitzarem la metodologia àgil **Scrum**[7] juntament amb una organització personal utilitzant taulells **kanban**[6].

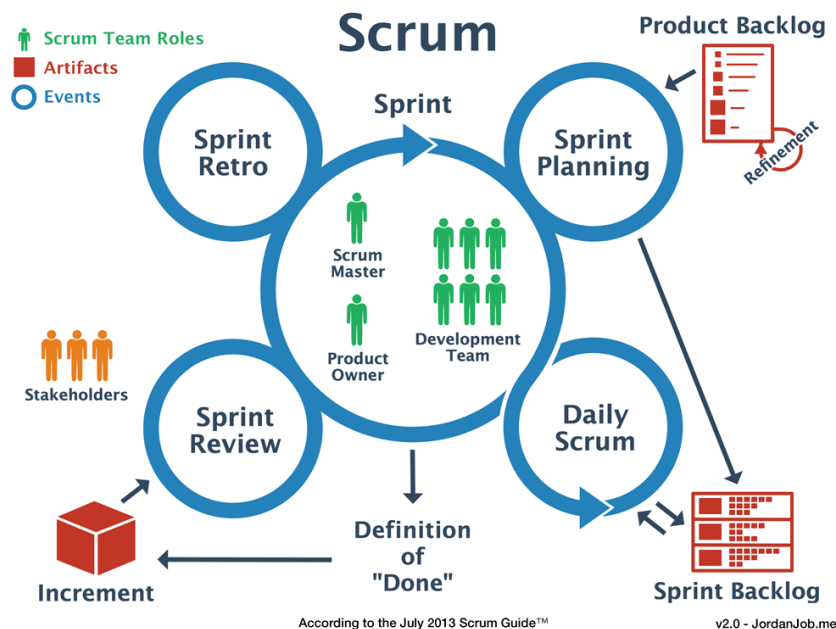
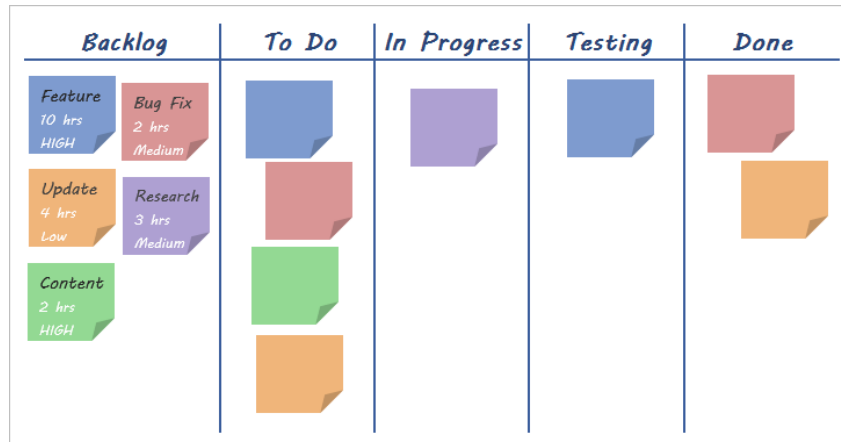


Figura 4.1: El mètode scrum gràficament.

Centrant-nos una mica més en la metodologia la implementació que nosaltres farem d'*scrum* serà una mica laxa substituint les reunions diàries per una reunió cada dimecres d'aproximadament una hora. Al acabar el mes es farà una *review meeting* (una reunió de repàs) on es parlarà de totes les tasques resoltes i no resoltes del *sprint* (feina a complir en un mes). Després d'aquesta reunió es programarà una reunió per als propers dies on es farà un anàlisi del *backlog*

Figura 4.2: Una pizarra *kanban*

(registre de feina pendent proposada) i s'inclouran al *sprint* les tasques que es considerin més importants.

Un cop assignat els *sprints* els desenvolupadors utilitzarem internament una pizarra de tipus *kanban* on apuntarem les tasques que hem de realitzar i el seu estat (pendent, preparada, en procés i feta) a part de subtasques necessàries per a poder resoldre l'*sprint*.

4.2 Mètodes de validació

Seguint el mètode *scrum* la nostra validació del projecte la farem a les reunions de final de mes (*review meeting*) on a més dels desenvolupadors hi assistiran l'equip directiu de la fundació. Aquí s'analitzaran els objectius assolits d'aquell mes i es prendran les mesures pertinents en cas de desviacions en la planificació (més seguiment, més recursos...).

D'altra banda la solució descrita incorpora sistemes de *testing* que permetran comprovar de forma molt més fàcil que el projecte compleixi els requisits donats.

Capítol 5

Recursos

5.1 Recursos personals

Aquí es disposarà d'un desenvolupador a 4 hores al dia (el realitzador del projecte), uns responsables que ajudaran en la validació del projecte i tasques d'organització i finalment la col·laboració esporàdica d'un altre desenvolupador. Aquest segon només col·laborarà unes 4 hores a la setmana aproximadament i assessorarà al desenvolupador principal.

5.2 Recursos materials

Disposem dels següents recursos materials:

1. Un ordinador portàtil: Un ordinador portàtil pcspecialist amb Debian Linux. Aquesta serà la principal eina de treball per a la realització del projecte.
2. Un cluster proxmox en un CPD: Aquest sistema serà necessari per muntar màquines virtuals de *testing* i per posar el sistema en producció.
3. Una oficina. Disposem d'un *coworking* a les Antigues Escoles de Gurb (Vic)

5.3 Recursos *software*

Disposem dels següents recursos no físics:

1. **Git i Github**: Git és una eina per a control de versions i el **GitHub**[3] és un servei públic de Git.[2]
2. **atom**: És l'editor de text patrocinat per **GitHub**. És molt versàtil i còmode, permet un gran nivell de personalització.[1]
3. Navegadors **Google Chrome i Firefox**: Ens permetran provar tota l'aplicació.
4. **Docker**. Es tracta d'un sistema de containerització de serveis (aplicacions).[8]

Capítol 6

Planificació temporal

6.1 Durada del projecte

Aquest projecte es va començar a gestar al voltant de l'any 2014. La fase de prototipatge va anar del 2014 a l'estiu del 2017 (Aquesta fase del projecte no s'estudia en aquest treball). Des del mes de juny de l'any 2017 fins a data d'avui el desenvolupament de software dins de la fundació Guifi.net ha anat a càrrec del autor d'aquest treball. L'inici del projecte plantejat data del mes de novembre de 2017. Totes les tasques exposades s'han acabat majoritàriament el mes d'octubre de 2018. Per tant aquesta planificació és una retrospectiva real del transcurs del projecte.

6.2 Tasques

| Tasca | Durada (hores) |
|--|----------------|
| Definició del projecte | 50 |
| 1. Planteig del projecte | 20 |
| 2. Aprenentatge eines | 30 |
| Gestió del projecte | 80 |
| 1. Definició del abast | 25 |
| 2. Planificació temporal | 15 |
| 3. Gestió econòmica i sostenible | 10 |
| 4. Plec de condicions | 10 |
| 5. Revisió preliminar | 10 |
| 6. Presentació resultats | 10 |
| Preparació d'entorns de desenvolupament | 30 |
| 1. Familiarització amb l'entorn Docker | 10 |
| 2. Implementació | 20 |

| Tasca | Durada (hores) |
|--|-----------------------|
| Actuacions sobre el <i>backend</i> | 260 |
| 1. Anàlisi de requisits | 20 |
| 2. Anàlisi canvi de plataforma | 20 |
| 3. Aprenentatge del llenguatge de la nova plataforma | 20 |
| 4. Implementació funcionalitats inicials amb la nova plataforma | 50 |
| 5. Implementació de noves funcionalitats | 80 |
| 6. Anàlisi sobre metodologies de <i>testing</i> | 10 |
| 7. Implementació dels <i>tests</i> | 20 |
| Actuacions sobre el <i>frontend</i> | 270 |
| 1. Anàlisi de requisits | 20 |
| 2. Anàlisi canvi de framework | 10 |
| 3. Aprenentatge del llenguatge del nou framework | 20 |
| 4. Implementació funcionalitats inicials amb el nou framework | 150 |
| 5. Implementació de noves funcionalitats | 70 |
| Actuacions sobre el model de dades | 90 |
| 1. Anàlisi del model de dades de Guifi.net | 20 |
| 2. Disseny de la integració del model de fibra al model actual (radio) | 20 |
| 3. Aprenentatge de la base relacional MySQL | 20 |
| 4. Implementació | 30 |
| Integració dels mapes | 60 |
| 1. Aprenentatge del software MapServer | 20 |
| 2. Importar les dades dels mapes de fibra al software MapServer | 40 |
| Posada a producció | 40 |
| 1. Disseny del procediment | 20 |
| 2. Implementació | 20 |
| Documentació | 130 |
| 1. Documentar <i>backend</i> | 20 |
| 2. Documentar <i>frontend</i> | 20 |
| 3. Preparar la memòria final | 70 |
| 4. Preparar presentació final | 20 |
| TOTAL | 1010 |

6.3 Dependències entre tasques

Es poden veure les dependències entre tasques al *Gantt* adjunt. En aquest treball la fase inicial encapçalada per **GEP** es trasllada a la fase final ja que el període de durada del projecte és previ a la realització del treball. Per tant el bloc de **GEP** anirà intercalat entre la finalització del projecte descrit i la redacció de la documentació.

Pel que fan les dependències sobre la realització del projecte, podem dir que el primer bloc, que consisteix en definir el projecte, és necessari per assolir qualsevol de les tasques posteriors. El següent bloc seria el de construir els entorns de desenvolupament. Aquests són necessaris per començar a desenvolupar sobre el projecte i per tant també precedeixen totes les implementacions posteriors.

El *backend* i el *frontend* no van acoplats. Es comencen en paral·lel aproximadament com es pot veure el *Gantt*,

Els canvis sobre el model de dades es poden fer en paral·lel a la implementació del *backend*.

La integració dels mapes depén exclusivament del *backend*.

Fins que no s'acaben les demés tasques no es pot procedir a dur a producció tot el sistema.

Capítol 7

Valoració d'alternatives i pla d'acció

7.1 Riscs

Al ser un projecte ja implementat en aquest apartat s'expliquen una mica els punts on hi ha hagut certes desviacions. En el gràfic de *Gantt* s'han marcat en vermell les 4 tasques on hi ha hagut incidències. Bàsicament es tracta de les 4 implementacions del software (*backend* i *frontend*) que eren les parts amb més durada i més complexes.

En el cas del *frontend* el problema trobat va ser la manca d'experiència prèvia en aquest tipus de solucions i el segon problema trobat ha estat el requeriment d'integrar el major part de codi del prototip al nou projecte.

Pel que fa la implementació del *backend* podriem dir que el major problema ha estat donat pels requeriments inicials i sobretot per les noves funcionalitats. Un punt a remarcar aquí seria la complexitat d'escriure codi de cara al servidor en un llenguatge asíncron com **JavaScript**.

7.2 Pla d'acció

Davant dels problemes descrits al apartat anterior i per minimitzar les desviacions causades es va procedir a fer el següent:

- *backend*: Aquí la forma d'avançar va consistir en demanar una auditoria del codi al desenvolupador auxiliar del projecte. D'aquesta manera jo em podia centrar a fer altres parts del projecte de forma simultània.
- *frontend*: Aquí la solució va passar per adaptar part del codi del prototip que s'havia d'integrar per a fer-lo compatible amb el nou *framework*. En alguns casos en veure la complexitat d'acoblar certes llibreries usades en el prototip es va optar per canviar-les per unes de similars i més ben adaptables.

Capítol 8

Implementació

En aquest capítol s'explicarà tot el procés d'implementació; es partirà del projecte en el seu inici i s'aniran exposant les tasques realitzades en cada component del sistema.

8.1 Entorns de desenvolupament

Per tal de poder treballar amb els múltiples serveis que conformen el projecte de forma còmode s'han creat uns entorns que faciliten la posada en marxa de tots els components que formen el sistema desenvolupat. S'ha utilitzat una tecnologia basada en contenidors.

8.1.1 La tecnologia Docker

Docker[8] és una tecnologia que permet encapsular aplicacions en unes estructures anomenades contenidors que contenen tots els elements necessaris per posar-les en marxa. Podriem dir que és un sistema de virtualització (entorns aïllats) de software, enlloc de ser entorns on es virtualitzi tot un sistema (Màquines virtuals, linux containers...). Aquests contenidors estan basats en imatges que internament són una extensió d'unes imatges base (normalment binaris de distribucions Linux). És una tecnologia molt emprada en sistemes basats en microserveis.[20]

Com podem veure a la figura 8.1 tenim uns contenidors basats en imatges, un registre (**registry**), repositori on es guarden les imatges i un client que gestiona aquestes instàncies.

Tal i com s'ha comentat prèviament les imatges es poden distribuir utilitzant un *registry*[16]. El *registry* és un servidor sense estat (stateless), altament escalable, capaç de guardar imatges **Docker** i distribuir-les fàcilment. Qualsevol usuari pot muntar el seu propi servidor, no obstant habitualment s'utilitza un *registry* públic mantingut directament per **Docker Inc.** anomenat **Docker Hub**.

Per tal de generar imatges necessitem un fitxer que actua com a plantilla (**Dockerfile**) i ens permet descriure com serà aquesta imatge:

- La imatge en la que es basarà (normalment una distribució de Linux)
- Les comandes i scripts que s'executaran per a crear-la

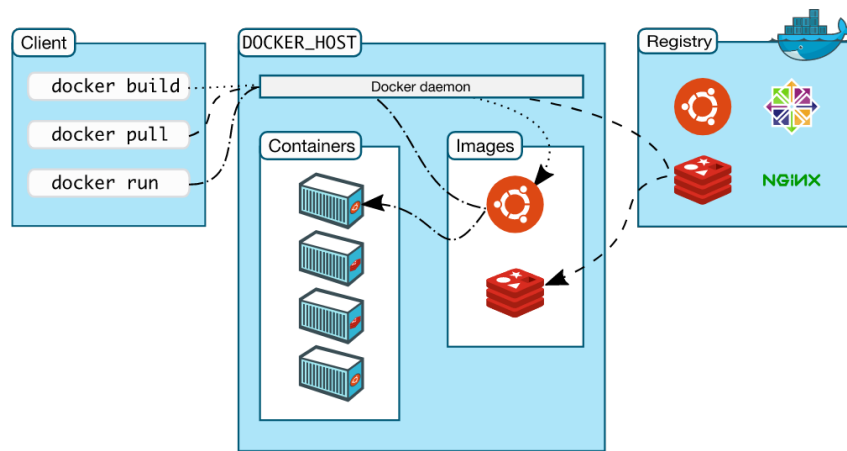


Figura 8.1: Arquitectura de la tecnologia Docker

- Els fitxers que se li copiaran
- El punt d'entrada: L'script que s'executarà primer en crear un contenidor.

Per defecte els fitxers creats dins d'un contenidor s'escriuen a la pròpia capa d'escriptura del contenidor (proporcionada pel host). Això implica que en el moment que es destrueix el contenidor la informació es perd. A més per la forma en que s'emmagatzemen les dades en el host, fa molt difícil la seva mobilitat. Per a solucionar aquesta problemàtica **Docker** proporciona dues alternatives:

- **Volums**: Es guarden en una part del sistema de fitxers gestionada directament per **Docker**. El seu ús està recomanat per **Docker** sobretot en entorns en producció.
- **Bind mounts**: En aquest cas són punts de muntatge situats directament a les carpetes d'usuari del host. És l'opció que fem servir en aquest projecte ja que a l'hora de desenvolupar és molt pràctic.

8.1.2 Els entorns dins de Guifi.net

En el cas del desenvolupament del fiberfy i dins del context de Guifi.net s'han desenvolupat dues imatges **Docker**.

docker-drupal-guifi

Aquesta és la primera imatge que es va desenvolupar. És anterior a l'inici del projecte però l'incloc a la memòria perquè el seu ús és necessari per a fer funcionar el sistema. És una imatge basada en **Debian jessie** que incorpora el següent software:

- Php 5.6.30

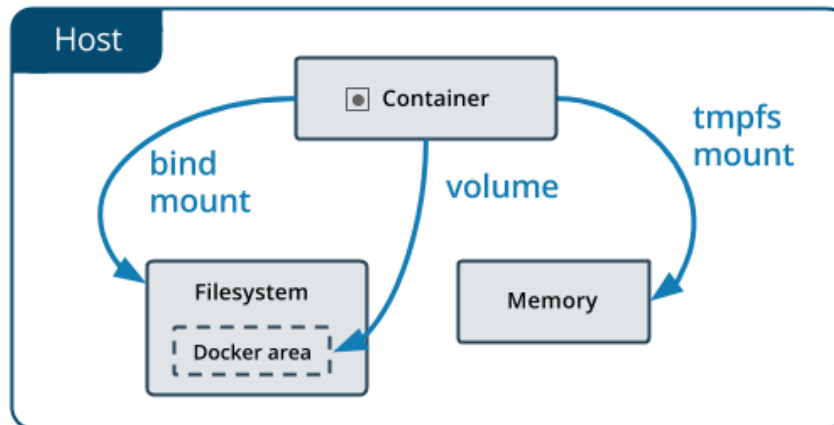


Figura 8.2: Tipus de punts de muntatge en Docker

- apache 2.4.10
- Drupal 6.38
- XDebug
- git
- wget
- paquets drupal-guifi i budgets

A part d'aquest conjunt de llibreries i binaris també incorpora scripts per posar en marxa una còpia de la web de Guifi.net

docker-fiberfy

Aquesta imatge s'ha fet per tal d'integrar el *backend* del fiberfy amb la resta de serveis necessaris per al sistema. És una imatge basada en **Debian stretch** que incorpora el següent software:

- node.js 8.12
- git
- curl
- wget
- sails 1.x
- waterline-auto

A part de contenir els anteriors paquets també conté un script en Perl que prepara l'entorn baixant-se el repositori git sobre la branca donada com a variable d'entorn. A l'apèndix D.1 s'adjunten el codi de tots els scripts utilitzats.

El funcionament dels contenidors basats en aquesta imatge és el següent:

1. S'executa un script d'entrada: *docker-entripoint.sh*. Aquest script simplement executa *user-mapping.sh*
2. *user-mapping.sh* és un petit script que es va extreure d'internet que serveix per canviar el UID i GID d'el username del contenidor. Aquests haurien de coincidir amb el user que executa el contenidor en el host. D'aquesta manera es permet compartir els fitxers generats en volums i treballar amb ells.
3. S'executa l'script *fiberfy-entry.pl*. Aquest és el principal script de configuració quan el volum encara no conté informació. La seva funcionalitat és baixar-se el codi de la branca donada (per defecte master) i instal·lar les dependències del backend.
4. Finalment el primer script executa la comanda donada al contenidor (normalment posarà el backend en funcionament en forma de dimoni)

docker-guifimaps

Aquesta imatge es va fer abans de començar aquest projecte, no obstant s'ha adaptat per a integrar-se correctament amb el *backend* del sistema com veurem més endavant. Basicament es tracta d'una imatge basada en **Debian stretch** que incorpora el software **MapServer**, un sistema per a publicació de dades geoespacial en el web (per a fer mapes)[19].

Composició

Per tal de treballar amb múltiples contenidors s'ha utilitzat una eina proveïda també pel projecte Docker que s'anomena **docker-compose**. Es tracta d'una petita aplicació que permet posar en marxa de forma conjunta diferents serveis basats en contenidors Docker de forma senzilla basant-se en una plantilla. Aquesta plantilla es pot generar fàcilment amb un fitxer escrit en **YAML** (Un subconjunt del llenguatge **JSON** amb una sintaxis més neta).

La composició que des de guifi.net fem servir per treballar amb el fiberfy segueix la següent arquitectura:

Com podem comprovar a la figura 8.3 la composició està formada per 5 contenidors:

- **docker-fiberfy**: fiberfy-server, explicat anteriorment.
- **docker-guifimaps**: guifimaps, explicat anteriorment.
- **mariadb**: És la base de dades que fem servir per al fiberfy-server (*backend*) i la web de guifi.net.
- **phpmyadmin**: És un gestor web per BBDD MySQL i MariaDB.

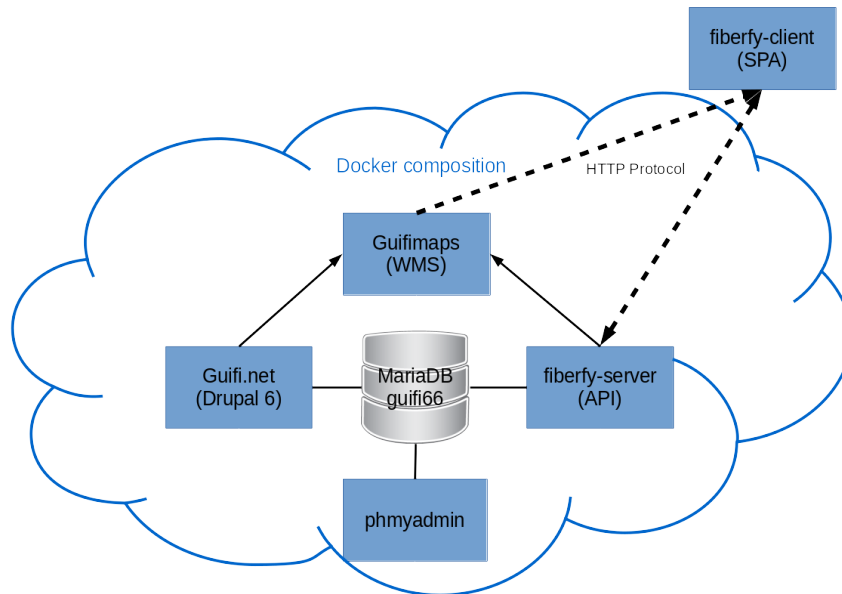


Figura 8.3: Arquitectura de la composició pel *fiberfy*

- **docker-drupal-guifi**: la web de guifi.net, explicat anteriorment.

A l'apèndix E.1 podem trobar el fitxer *docker-compose.yaml* d'aquesta composició.

8.2 Backend (fiberfy-server)

8.2.1 Contextualització

L'eina de gestió d'infraestructures de fibra òptica, d'ara en endavant, *fiberfy* està formada per un *backend*: *fiberfy-server* basat en una API **REST** i escrit en **JavaScript**. Per tal d'executar codi en JavaScript a la banda del servidor es fa servir l'entorn d'execució *node.js*.

Al moment de començar el projecte, novembre de 2017, existia un prototip de l'eina que tenia les següents problemàtiques:

- Feia servir un *framework* amb un suport molt pobre per part dels desenvolupadors. La documentació era escassa.
- La base de dades usada era **SQLite**, una base de dades relacional en forma de fitxer força simple, molt útil però poc escalable. Des de Guifi.net ens interessava integrar la base de dades **MySQL** existent en aquell moment amb la del *fiberfy*.
- Harmonització del model de dades del *fiberfy* amb el model actual de Guifi.net (Nodes, zones...)
- Només es permetia que un projecte fos editat per un únic usuari. Això en un entorn multioperador com el nostre era un problema important.

- No es podien importar projectes des de cap font externa. Molts operadors abans de que es desenvolupés aquesta eina treballaven amb eines com **Google Earth** o eines de **CAD**.
- L'aplicació no era capaç de calcular les distàncies entre les localitzacions donades.
- No suportava **LDAP** com a agent autenticador.

El codi font de l'aplicació es pot trobar a: <https://github.com/guifi/fiberfy-server/>.

8.2.2 Canvi de framework

A l'inici del projecte, seguint el requeriment d'unificar el model de dades de l'eina de fibra amb el de Guifi.net, es necessitava canviar la base de dades SQLite per la base de dades mariaDB. El framework trails.js és un fork de sails.js que pretenia ser una millora d'aquest. No obstant quan es va intentar un canvi tant simple com el de canviar un motor de dades per un altre es va veure que no existia la documentació adequada per fer el canvi. Un altre aspecte força preocupant ha estat veure la poca activitat que presentaven els seus repositoris, un indicatiu força clar d'abandonament del projecte. Veient que en canvi el projecte de sails.js presenta una forta activitat i que es trobava en constant evolució, es decideix canviar un framework per l'altre.

El framework sails.js segueix l'arquitectura del software **MVC**, que implica segmentar les parts lògiques de l'aplicació en:

- **Persistència(Model):** Accés a les dades emmagatzemades.
- **Vistes:** La interacció amb l'usuari.
- **Controladors:** El Business logic de l'aplicació.

A la figura 8.4 es pot veure l'esquema de **MVC**.

A la figura 8.5 es pot veure l'estructura de directoris del *backend*.

8.2.3 Canvi de base de dades

Tenint com a objectiu la integració del model de dades de Guifi.net amb el model imperant al *fiberfy* el primer pas a seguir era el d'unificar la mateixa base de dades. El prototip del *fiberfy* feia servir una base de dades força simple utilitzant SQLite, un petit SGBD relacional molt lleuger que utilitza un únic fitxer a disc per emmagatzemar les dades i que es pot integrar en el software en forma de llibreria[22].

Cal remarcar que canviar de motor de dades usat pel *backend* és molt fàcil ja que sails.js utilitza un **ORM** anomenat *Waterline*. Un **ORM**[11] és una llibreria que permet associar entitats de dades com a objectes (en llenguatges orientats a objectes), d'aquesta manera es permet interactuar amb la base de dades utilitzant el propi llenguatge de l'aplicació (no el de la base de dades).

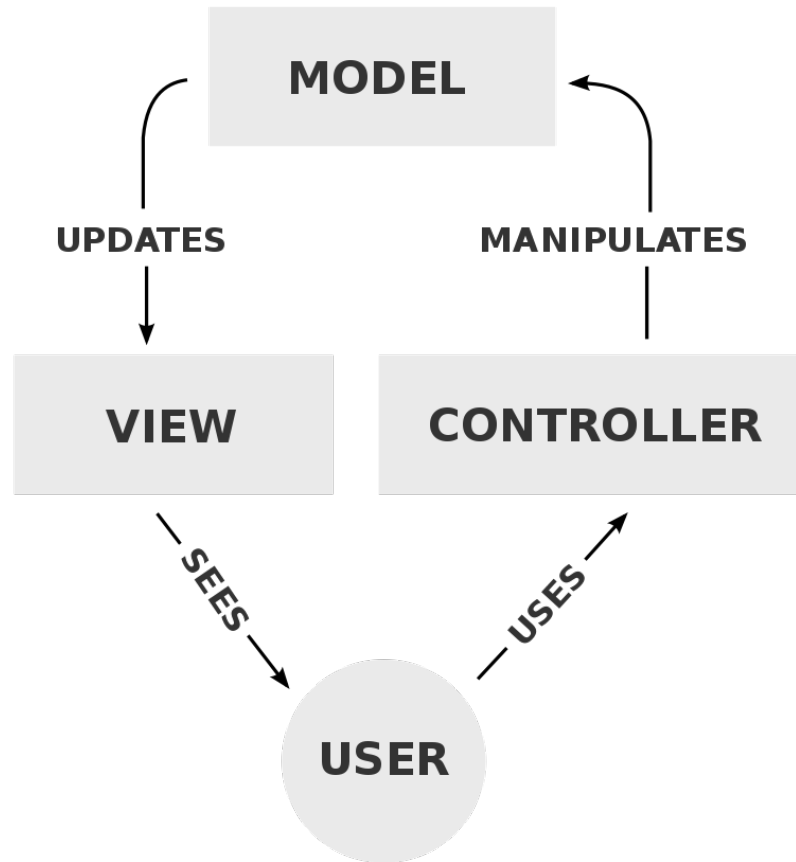


Figura 8.4: Arquitectura MVC

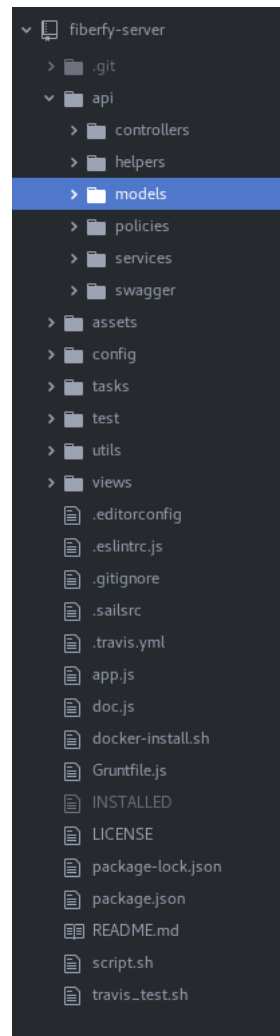
Model de dades

Per tal d'integrar les dades dels dos sistemes (web de Guifi.net i *fiberfy*) s'han fet les següents modificacions:

- S'han canviat els noms dels camps de l'entitat *Site* per adaptar-la a la taula homònima de Guifi.net *guifi_location*. A la figura 8.6 podem veure quins canvis s'han fet.
- S'ha afegit el model de zones existent a Guifi.net. A la figura 8.7 es poden veure quins camps conté.
- S'ha segmentat el model *Fiber* inicial en: *Cable*, *Tube* i *Fiber*
- S'ha afegit l'entitat *ProjectOwnership* que fa de join entre usuaris i projectes; d'aquesta manera podem assignar més d'un usuari per projecte.

A l'apèndix F.1 es poden consultar tots els models declarats.

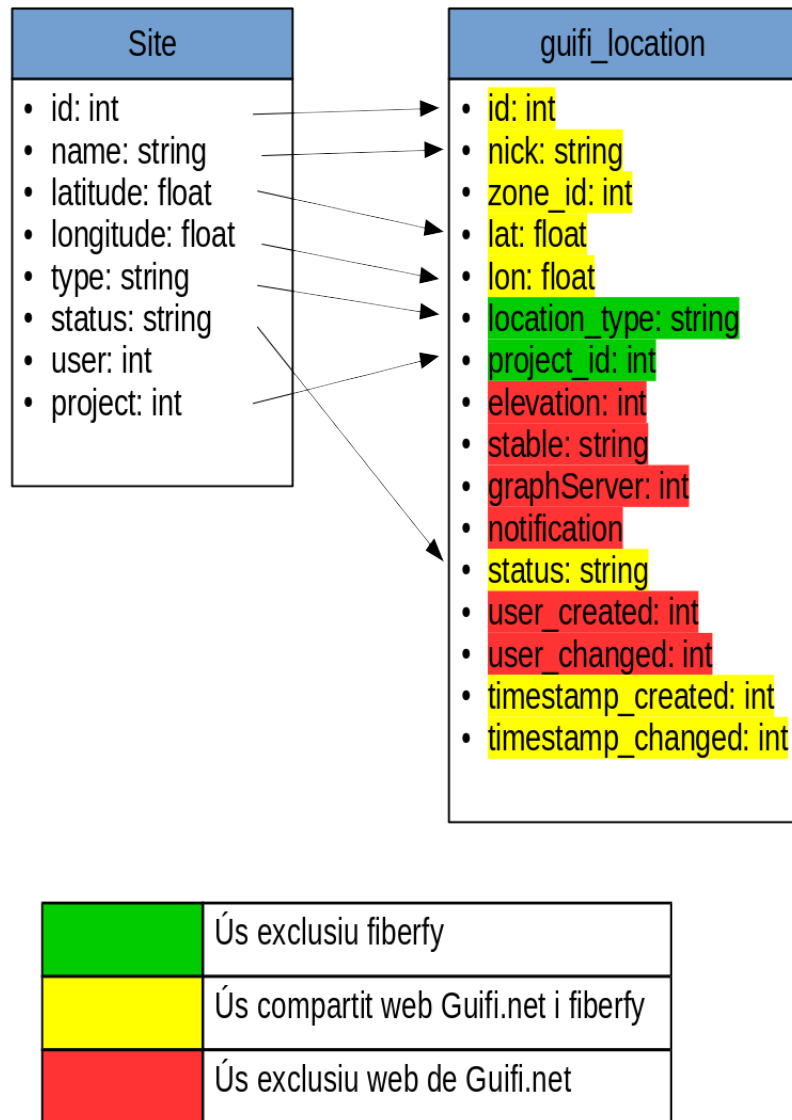
A l'apèndix C es pot observar un diagrama ER amb tots els models i relacions.

Figura 8.5: Estructura de directoris de *fiberfy-server*

8.2.4 Integració amb LDAP

LDAP és un conjunt de protocols usat per accedir a dades guardades de forma centralitzada a través de la xarxa. Es basa en l'estàndard X.500 per a compartir directoris, però és menys complex i lleuger en termes de consum de recursos[17]. **LDAP** es fa servir habitualment per mantenir directoris d'usuari dins de les organitzacions a més de permetre assignar atributs personalitzats a cada usuari. Gràcies a aquestes capacitats **LDAP** ens permet assignar permisos per usuari molt útil a l'hora d'integrar diferents sistemes.

Els directoris **LDAP** presenten una estructura en forma d'arbre[23]; la seva arrel queda descrita pels *tags dc* (en forma de domini) com per exemple: *dc=guifi*, *dc=net*. Al següent nivell es poden observar les unitats organitzatives *ou*; per exemple podria ser un departament: *ou=developers*. Finalment els usuaris es troben assignats en un determinat departament, s'utilitzen les sigles *sn* o *cn*. Un exemple seria *cn=roger.garcia*. A la figura 8.8 es pot veure un

Figura 8.6: Canvi de l'entitat *Site*

exemple de directori.

En aquest projecte nosaltres partim de l'existència d'un servei de **LDAP** en producció que té la següent estructura:

- L'arrel és: *dc=guifi, dc=net*
- Hi ha 3 unitats organitzatives, no obstant nosaltres només ens centrarem en *ou=glirusers*
- Finalment aquí hi trobem els usuaris del **GLIR** (operadors de Guifi.net) i també els grups de permisos. En el cas del *fiberfy* només necessiten estar associats al *cn=lectura*.

| | | |
|--|--------------------------------------|--|
| | Ús exclusiu fiberfy | |
| | Ús compartit web Guifi.net i fiberfy | |
| | Ús exclusiu web de Guifi.net | |

| guifi_zone |
|---|
| <ul style="list-style-type: none"> • id: int • title: string • nick: string • body: string • master: int • time_zone: string • dns_servers: string • ntp_servers: string • graph_server: string • homepage: string • notification: string • ospf_zone: string • minx: float • maxx: float • miny: float • maxy: float • local: string • nodexchange_url: string • refresh: int • remote_server_id: int • weight: int • user_created: int • user_changed: int • timestamp_created: int • timestamp_changed: int |

Figura 8.7: Integració de l'entitat *Zone*

Per tal de poder treballar amb directoris **LDAP** s'ha integrat un mòdul de node.js com a servei de sails.js. És necessària a més la instal·lació d'un mòdul de node.js que tingui suport per ldap. El *fiberfy-server* fa servir un modul força senzill que s'anomena ldap.js. Bàsicament el codi del servei es divideix en dues funcions principals. Per una banda la primera serveix per consultar l'existència d'un usuari al directori. La segona en canvi permet l'autenticació de l'usuari mitjançant el password proporcionat. A partir d'aquí hi ha diferents callbacks que permeten controlar el comportament de l'aplicació, fluxe d'execució, gestió d'errors i de resultats. A l'apèndix F.2 es troba el codi de LDAP.

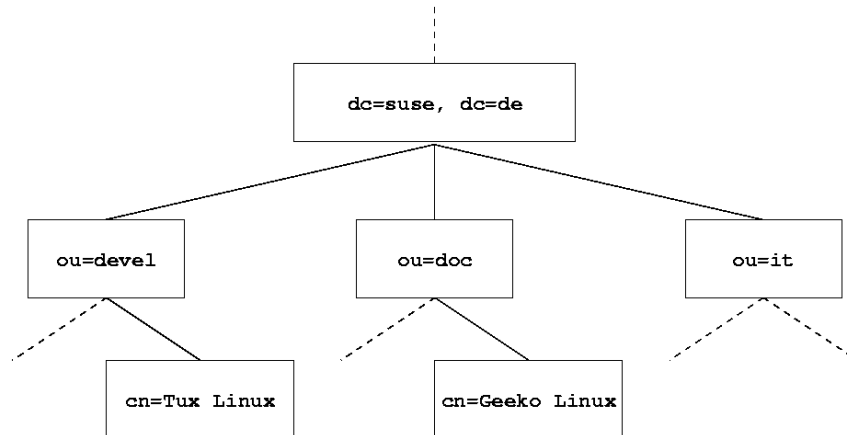


Figura 8.8: Exemple de directori LDAP

8.2.5 Implementació de l'importador d'infraestructura

L'importador d'infraestructura no és més que la funcionalitat que permet importar mapes de punts i línies en un format **GIS** reconegut sobre el model de dades del *fiberfy*. Actualment estan suportats els formats de Google Maps, **KMZ** i **KML** i el format **GeoJSON** (basat íntegrament en JSON).

Una de les característiques més interessants d'aquest algoritme és que permet relacionar les línies d'un mapa que es troben molt a prop. Això és molt útil ja que el nostre model de dades es basa en aquestes relacions i els estàndards **GIS** simplement permeten mostrar dades geomètriques però no les relacionen.

El funcionament de l'algoritme és el següent:

1. Es validen els paràmetres que li arriben a l'API.
2. Si el format de les dades es difereix de GeoJSON es transforma en aquest format.
3. S'iteren tots els elements compatibles de l'estàndard GeoJSON: Punts (Points) i línies (LineStrings) i s'emmagatzemen en dues arrays.
4. S'iteren les línies i per a cada una d'elles es busca coincidència amb un punt existent: si existeix no se'n crearà un de nou i es relacionarà el tram de fibra amb aquell punt, en cas contrari es crearà un punt nou i s'unirà el tram.¹
5. S'importen els Sites (Llocs) a la base de dades de forma asíncrona, utilitzant l'ORM.
6. S'importen els Paths (Trams) a la base de dades de forma asíncrona, utilitzant l'ORM.

Es pot consultar el codi a l'apèndix F.4.

¹D'aquesta manera es permet ajuntar les línies que es troben juntes amb un marge de tolerància donat per l'usuari.

8.2.6 API REST

REST[14] és una arquitectura de programari pensada per a sistemes distribuïts basats en hipermèdia com per exemple la web (protocol http). Els serveis **REST** són centrats en les dades; el seu ús principal és el d'actuar com a interfície entre un *backend* i un *frontend*. Una **API** per ser considerada **REST** ha de complir les següents condicions:

1. **Uniform interface:** Bàsicament vol dir que s'identifica clarament l'interacció amb els recursos incloent tota la informació necessària per processar les dades.
2. **Client-servidor:** Separació entre *backend* i *frontend* de la mateixa manera que aquest projecte.
3. **Stateless:** Cada missatge conté la informació necessària per processar la petició.
4. **Cacheable:** Les respostes del servidor poden guardar-se en una memòria cache per tal d'agilitzar el sistema.
5. **Layered system:** El client no assumeix que hi ha una connexió directa amb el servidor final. Poden existir sistemes software o hardware entre ells.

Tal i com s'ha comentat anteriorment el sistema *fiberfy* es basa íntegrament en una API d'aquestes característiques. Per defecte el framework usat per desenvolupar el *backend* incorpora una característica que ens automatitza la feina per construir l'API: els *Blueprints*. Això no és més que un mòdul que automatitza la construcció d' APIs basades en **CRUD**. Bàsicament per utilitzar aquesta característica hem de declarar un Controlador (seguint **MVC**) amb el mateix nom que un Model acabat en Controller. Si aquest està buit sense cap altre acció declarada simplement s'exposarà el **CRUD** generat automàticament. És possible canviar el comportament d'una acció Blueprint declarant una acció al controlador amb el mateix nom que l'acció a substituir.

Les accions Blueprint utilitzades en aquesta aplicació són les següents:

- *find*: És l'acció que ens permet fer queries. La seva **URI** té el següent format: GET `http(s)://host:port/api/model` (amb els paràmetres de búsqueda pertinents)
- *findOne*: És l'acció encarregada de retornar una entitat d'un model. Presenta el següent format: GET `http(s)://host:port/api/model/:id`
- *update*: És l'acció encarregada d'actualitzar el contingut d'una entitat. Presenta el següent format: PATCH `http(s)://host:port/api/model/:id` (l'entitat actualitzada com a body)
- *create*: És l'acció encarregada de crear una nova entitat. Presenta el següent format: POST `http(s)://host:port/api/model/` (la nova entitat com a body)
- *destroy*: És l'acció encarregada d'eliminar una entitat. Presenta el següent format: DELETE `http(s)://host:port/api/model/:id`

- *add*: Aquesta acció ens permet lligar una entitat amb una altra a través d'algun atribut compatible (clau externa). Presenta el següent format: PUT http(s)://host:port/ api/model/:id/association/:fk
- *remove*: Aquesta acció ens permet deslligar una entitat amb una altra a través d'algun atribut compatible (clau externa). Presenta el següent format: DELETE http(s)://host:port/ api/model/:id/association/:fk

A la documentació del projecte sails.js es pot trobar tota la informació referent als Blueprints: <https://sailsjs.com/documentation/reference/blueprint-api>.

Documentació API

Quan un projecte té vocació de ser utilitzat com a part d'un entorn multiservei és necessari que la seva interfície estigui ben documentada. En el cas d'una API Rest això inclou detallar com és, quins mètodes suporta i com han de ser els paràmetres d'aquests. Per tal de realitzar aquesta feina de forma còmode i eficaç s'ha utilitzat una eina documental: **Swagger**.

Swagger[21] són un conjunt d'eines per tal de dissenyar, implementar i visualitzar APIs Rest. El llenguatge per especificar les APIs està basat en YAML o JSON.

Bàsicament en aquest projecte només s'ha utilitzat l'especificació de l'API utilitzant YAML, JSON i el visualitzador de l'especificació: Swagger UI.

El Swagger UI s'ha integrat al *fiberfy-server* de tal forma que l'especificació de l'API es pot consultar accedint directament des d'un navegador a la URI: http(s)://host:port/api-doc

A la figura 8.9 es pot observar l'aspecte del visor Swagger (Swagger UI).

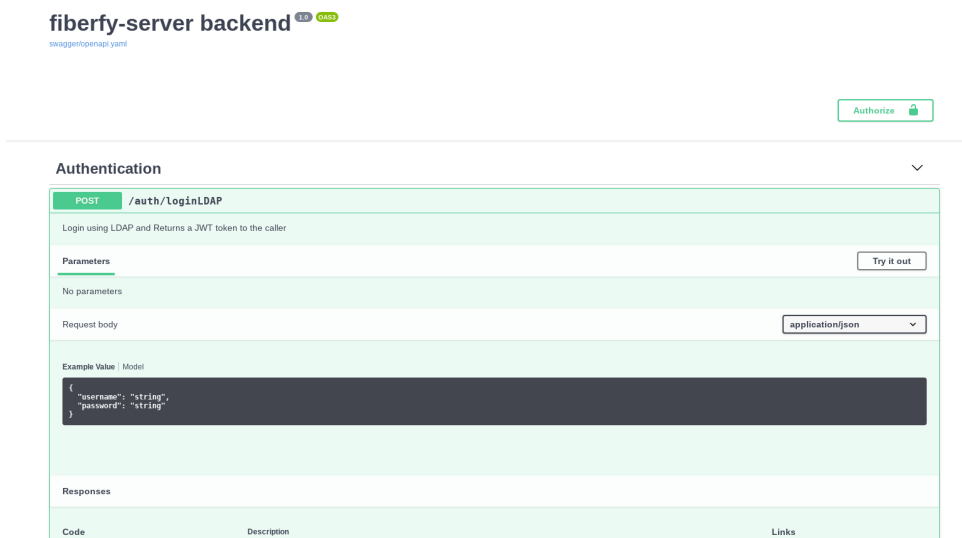


Figura 8.9: Swagger UI mostrant API *fiberfy-server*

D'altra banda s'ha desenvolupat un petit script en JavaScript que a partir d'unes metadades incloses a la declaració de models (disponibles a l'apèndix F.1) és capaç d'incloure'ls a l'especificació de l'api. Aquest script es troba a l'apèndix F.3.

8.3 Frontend (fiberfy-nt)

8.3.1 Contextualització

Tal i com s'ha exposat a la secció 8.2.1 el *fiberfy* es compon d'un *frontend*. Inicialment el *frontend* venia acoplat directament al *fiberfy-server* en forma de dues vistes en forma de **SPA**. En el moment de començar el projecte tenia les següents limitacions:

- S'havia desenvolupat utilitzant **jQuery**, una llibreria molt útil per a la manipulació de l'arbre **DOM** i capturar esdeveniments però poc còmode a l'hora de desenvolupar grans aplicacions basades en una única pàgina.
- No es podia tirar més d'una mànega de fibres per cada tram (canalització, aeri).
- Molts problemes en mantenir l'estat i la reactivitat. Això vol dir que alguns canvis no es manipulaven correctament.
- Dificultat en mantenir el codi degut a la pròpia naturalesa del software escrit en **jQuery**

Degut a tots els problemes plantejats anteriorment durant el desembre de 2017 es va plantejar reescriure l'aplicació utilitzant un *framework* específic per a la construcció d'aquest tipus d'aplicacions.

Inicialment es va plantejar l'opció d'utilitzar el *framework* **Angular.io** desenvolupat per **Google Inc.** ja que és una de les opcions més conegudes. No obstant es va desestimar pels següents inconvenients:

- Ecosistema molt tancat. Després de fer alguns experiments utilitzant la llibreria es va comprovar que era pràcticament impossible integrar de forma còmode i fiable part del codi existent.
- La documentació està escrita per usar **TypeScript**, un dialecte de **JavaScript** que incorpora la declaració fortament tipada de variables. Això implica aprendre un nou llenguatge i encara complica més la integració de codi existent.

En aquell moment i pràcticament per casualitat vam topiar amb un nou *framework* que feia èmfasi en l'adopció progressiva de la llibreria: **Vue.js**.

El codi font de l'aplicació es pot trobar a: <https://github.com/guifi/fiberfy-nt/>.

A l'apèndix A es pot trobar el manual d'usuari de l'eina. Inclou tots els formularis i s'explica l'interacció amb l'usuari.

8.3.2 Nova arquitectura

Introducció a Vue.js

Vue.js[26] és un framework JavaScript per a *frontend* molt enfocat a l'adopció progressiva de la llibreria. Els seus autors entenen com a progressiu, un framework molt modular que el



(a) Logotip de jQuery



(b) Logotip de Vue.js

Figura 8.10: Logotips de llibreries *frontend*

programador pugui anar adoptant segons les demandes del seu projecte i a més que sigui capaç d'interactuar amb altres llibreries. De fet és el framework ideal per aquest projecte ja que un dels objectius a complir era el de re-aprofitament del prototip: veure 2.2.

L'arquitectura bàsica de Vue.js es basa en la creació d'una instància Vue (normalment en SPA només hi ha una única instància) que conté en el seu interior uns components. Aquests són els encarregats d'interactuar amb l'usuari i mostrar informació. Bàsicament un component de Vue.js és un petit fragment de l'aplicació que desenvolupa una funcionalitat concreta, d'aquesta manera es poden segmentar clarament les parts del sistema permetent la construcció de serveis escalables i sobretot mantenibles en el temps.

Un component conté les següents parts clarament diferenciades:

- **Template:** És el fragment de codi escrit en pseudo HTML que serveix per renderitzar el component. Inclou les assignacions de variables i tot el que implica la visualització del mateix.
- **Script:** És la part que expressa el comportament del component. Es troba escrit en JavaScript i incorpora els mètodes, les dades (variables, objectes...) i la comunicació del component.
- **Style (opcional):** És el fragment que expressa els estils de la Template; per defecte es fa servir CSS.

Vuex un gestor d'estats

En una aplicació SPA un gestor d'estats és aquell mòdul encarregat de gestionar la persistència (les dades amb les que treballa l'aplicació) a nivell de *frontend*. Això inclou per exemple en el cas del *fiberfy* les arquetes que es mostren al mapa o els projectes de l'usuari etc. També és la part del sistema encarregada de treballar amb aquestes dades (introduir-les, modificar-les, donar-les...) a més en el cas d'aquesta aplicació també és la part del codi que s'encarrega de

transferir les dades al backend utilitzant l'API REST. El projecte Vue.js conté un mòdul de gestió d'estats que es pot integrar molt fàcilment a qualsevol aplicació Vue: el **Vuex**

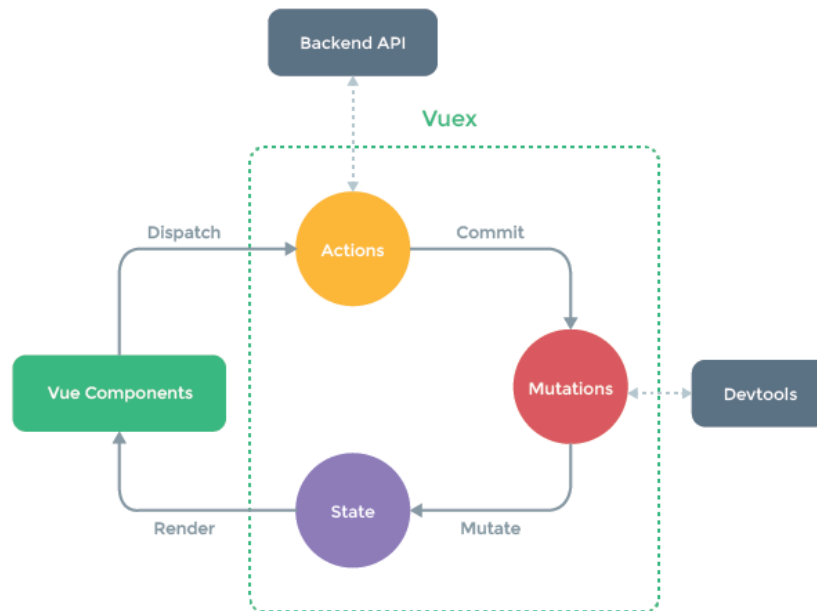


Figura 8.11: Esquema de funcionament Vuex

Aleshores els components per tal d'obtenir les dades del sistema primer de tot interactuen amb el gestor d'estats **Vuex**. A la figura 8.11 es pot observar com interactuen els components amb el gestor d'estats.

8.3.3 Llibreries utilitzades

Una de les claus d'haver utilitzat el framework Vue.js era la possibilitat d'integrar llibreries existents de forma còmode. A la següent llista es mostren les principals llibreries utilitzades i quin ús se'ls hi ha donat:

- **leaflet.js**: [25] És una llibreria per treballar amb mapes raster. És una llibreria OpenSource que fa molt èmfasi en la lleugeresa i la documentació senzilla. És la llibreria amb la que s'havia programat els mapes del prototip. S'ha reaprofitat al màxim el codi del prototip.
- **fabrik.js**: És una llibreria per facilitar l'ús de canvas. És una llibreria que permet crear polígons geomètrics i interactuar entre ells. S'ha fet servir per generar els gràfics de fusions. Es pot veure un exemple al manual de l'aplicació que es troba a l'apèndix A.
- **Bootstrap**: És una llibreria gràfica basada en CSS feta per Twitter. Permet crear interfícies d'usuari web força agradables i de forma senzilla.

8.4 Servei de mapes (guifimaps)

8.4.1 Contextualització

Es tracta d'uns petits scripts que s'encarreguen de preparar les dades vectorials per tal de crear mapes utilitzant el protocol **WMS**. Per tal d'exposar els mapes en format **WMS** es fa servir el software MapServer. Es tracta d'una aplicació que permet processar mapes vectorials i extreure'n imatges raster per tal d'estalviar el comput als dispositius receptors²

8.4.2 Millores aplicades

Tot-hi que el desenvolupament d'aquesta eina no ha variat massa, s'han ampliat els scripts de càrrega de mapes per tal d'importar les dades geogràfiques de presència de fibra òptica generades amb el fiberfy-server.

S'ha utilitzat l'API Rest del fiberfy per establir la comunicació entre l'script i el backend.

El fluxe de funcionament del servei amb els canvis aplicats és el següent:

1. S'executa el cron cada cert temps per actualitzar els mapes
2. El cron crida a un primer script en PHP.
3. L'script en PHP consulta les dades de la web de Guifi.net (nodes sense fils). Després s'autentica a l'API del fiberfy amb unes credencials. Acte seguit es baixa els mapes de fibra òptica (Llocs i Trams) i ho volca en un fitxer GML.
4. El cron executa una eina que transforma el GML en un fitxer ESRI (Un altre estandard GIS que ofereix més rendiment).

Aleshores Mapserver ja té els fitxers de mapa actualitzat i pot respondre les peticions dels usuaris en WMS.

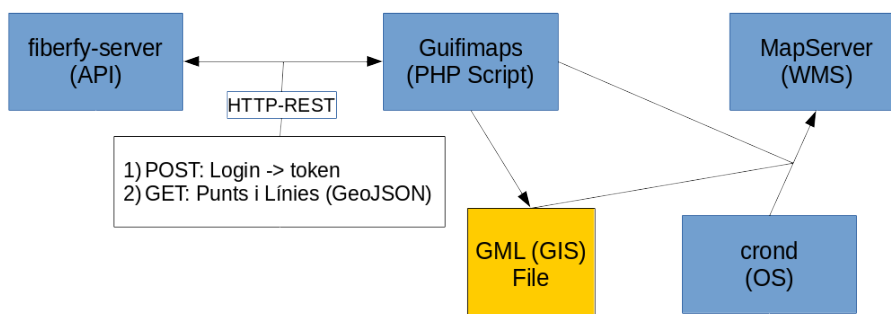


Figura 8.12: Esquema de funcionament Guifimaps amb les millores

²A dia d'avui gràcies a l'augment de comput de càlcul ja no és tant necessari el processament de mapes en servidors. Tecnologies com WebGL faciliten la representació de mapes vectorials als clients.

Capítol 9

Posada en marxa

Al capítol 8 s’ha parlat de l’arquitectura del software *fiberfy* i les eines que l’envolten, en aquest capítol es mostrarà com es posa en marxa el sistema sobre entorns de proves i sobre entorns de producció.

9.1 Entorns físics

9.1.1 Testing

Per tal de fer totes les proves s’ha fet servir un ordinador preparat com a servidor amb les següents característiques:

- **Processador:** Core 2 Quad Q8400 2.66 GHz (Quad core)
- **Memòria:** 16 GB RAM DDR3
- **Discs durs:** 2 x 500 GB SATA (RAID 1 per Hardware)
- **Controladora storage:** LSI Logic SAS1064ET

9.1.2 Producció

S’ha fet servir un servidor allotjat al **CPD** de la Fundació Guifi.net.

9.2 Plataforma de virtualització

A dia d’avui quasi ningú utilitza entorns *bare-metal*¹ per córrer serveis, exceptuant alguns sistemes d’alt rendiment com per exemple bases de dades d’entitats financeres.

¹Són entorns clàssics on una màquina física executa un sistema operatiu amb serveis, no hi ha quasi aïllament.

9.2.1 Introducció a la virtualització

Abans d'entrar en detall sobre els sistemes de virtualització s'exposarà breument en que consisteix la virtualització de sistemes.

Hi ha diferents tipus de virtualització, en aquest apartat s'exposa la virtualització completa de sistemes operatius accelerada per hardware (*full virtualization*). Bàsicament aquest tipus de virtualització serveix per crear *ordinadors virtuals* que poden executar un sistema operatiu de la mateixa manera que ho faria un ordinador físic estàndard.

El concepte de virtualització és força fàcil d'entendre no obstant la seva implementació no és tant trivial. Per executar virtualització completa són necessaris els següents requisits:

- Suport d'extensions de virtualització: Bàsicament es tracta d'unes extensions del processador que permeten treballar amb espais virtuals de memòria (incorpora una **MMU** per gestionar l'accés a memòria física de les màquines virtuals). També incorpora suport per treballar amb diferents nivells de privilegi basats en anells (això ja s'utilitzava per executar el kernel dels sistemes operatius i les aplicacions d'usuari en un nivell més baix de privilegi).
- Presència d'un *hypervisor*. Un *hypervisor* és el software encarregat de gestionar l'execució de múltiples màquines virtuals en un host; gestiona l'assignació de temps de **CPU**, accés a la memòria, als recursos del host (perifèrics, xarxa...)

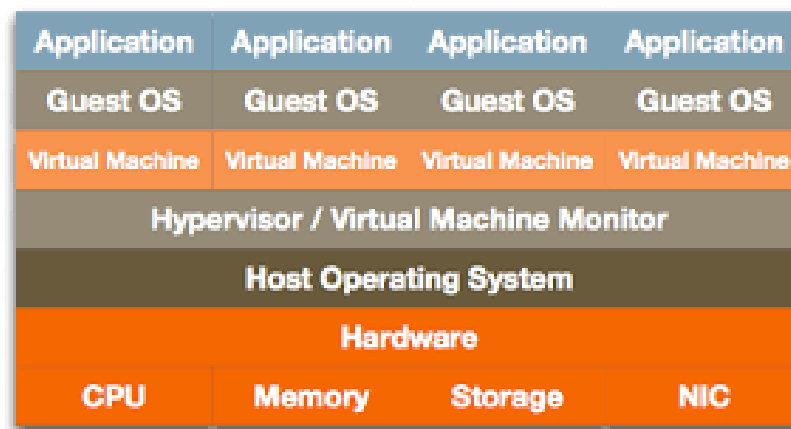


Figura 9.1: Pila de sistemes en virtualització completa

9.2.2 Anàlisi de plataformes

Tot-hi que hi ha hypervisors que es poden instal·lar com una aplicació d'usuari, per exemple VirtualBox, a l'hora de mantenir servidors en producció o testing no són els més adequats. Per tant aquí s'han analitzat plataformes de virtualització senceres pensades per sistemes 24/7. A la següent taula podem veure una comparativa dels 3 sistemes més interessants observats:

| | VMWare vSphere | Citrix XenServer | Proxmox |
|----------------------------|-----------------------|-------------------------|----------------------|
| License | Propietary | GPLv2 | GPLv3 |
| Hypervisor | VMWare ESXi | Xen | KVM |
| Full virtualization | Si | Si | Si |
| Clustering | Si | Si | Si |
| Paravirtualization | No | Si | No |
| Container support | No | No | Si (LXC) |
| Cost | 939,50 € (Standard) | Gratis(Sense suport) | Gratis(Sense suport) |

S'ha triat Proxmox VE ja que els entorns de Guifi.net estaven tots basats en aquest software i a més es considera una funcionalitat molt interessant el poder crear contenidors de sistema basats en LXC.

9.3 Entorns virtuals

En aquesta secció s'exposarà com són les instàncies virtuals de producció i de testing. La màquina virtual és idèntica en ambdós casos (mateix sistema operatiu, memòria, software...) per tal de mantenir al màxim la correspondència.

La subsecció 9.3.1 mostra com funciona la creació d'instàncies en un cluster de Proxmox VE.

9.3.1 Creació de la instància (testing)

S'ha triat la creació d'una màquina virtual completa i no un contenidor LXC perquè no es recomana executar contenidors basats en **Docker** sobre contenidors LXC per seguretat².

Aleshores per començar hauríem d'accedir al host que conté la plataforma Proxmox:
<https://10.90.230.34:8006>

²Docker pot accedir com a root al kernel del sistema host i això podria provocar un accés a tot el sistema en cas de patir un atac al contenidor.

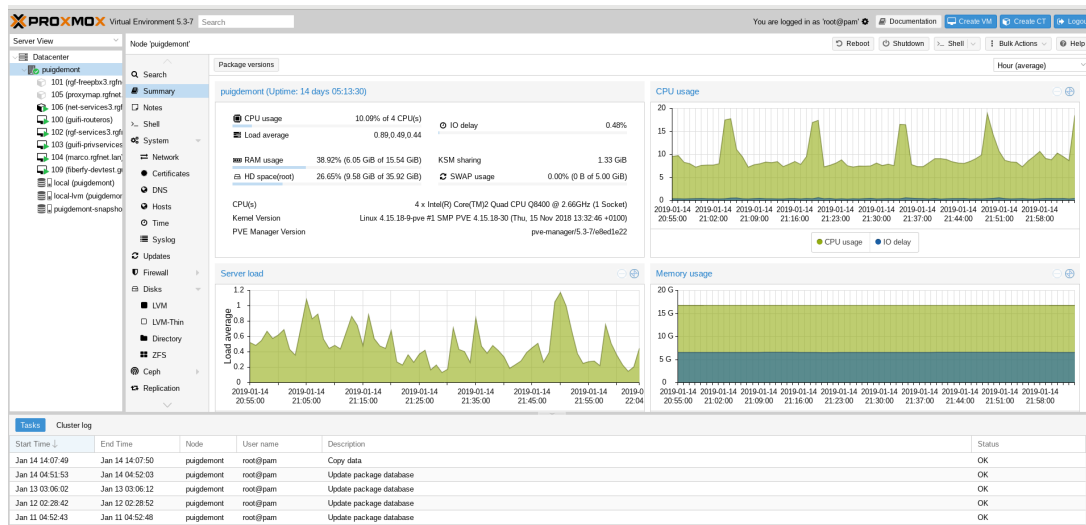


Figura 9.2: Dashboard de la plataforma Proxmox VE

Després apareixerà un menú lateral a l'esquerra amb tots els nodes del cluster, les seves màquines virtuals i contenidors. Per crear una màquina virtual nova s'haurà prémer el botó dret de l'apuntador sobre el node sobre el qual volem crear la instància: l'únic node de testing porta per nom *pauigdemont*.

Aleshores es mostrarà un desplegable on s'haurà de prémer l'opció de *Create VM*. Acte seguit apareixerà un formulari on s'hauran d'especificar les opcions de la màquina virtual: nom, id, memòria, cores assignats, storage (partició sobre LVM Thin), CD d'instal·lació del sistema i xarxa.

The screenshot shows the 'Create: Virtual Machine' dialog box with the 'General' tab selected. The fields are as follows:

| Field | Value |
|----------------------|--------------------------|
| Node | puigdemont |
| VM ID | 107 |
| Name | fiberfy-devtest |
| Resource Pool | |
| Start at boot | <input type="checkbox"/> |
| Qemu Agent | <input type="checkbox"/> |
| Start/Shutdown order | any |
| Startup delay | default |
| Shutdown timeout | default |

At the bottom, there is a 'Help' button, an 'Advanced' checkbox (checked), and 'Back' and 'Next' buttons.

Figura 9.3: Formulari de creació de màquines virtuals (1)

The screenshot shows the 'Create: Virtual Machine' dialog box with the 'Confirm' tab selected. It displays a table of configuration keys and values:

| Key ↑ | Value |
|----------|--|
| balloon | 1024 |
| cores | 2 |
| ide2 | local:iso/debian-9.6.0-amd64-netinst.iso,media=cdrom |
| memory | 6144 |
| name | fiberfy-devtest |
| net0 | virtio,bridge=vmb1 |
| nodename | puigdemont |
| numa | 0 |
| ostype | i26 |
| scsi0 | local-lvm:20 |
| scsihw | virtio-scsi-pci |
| sockets | 1 |
| vmid | 107 |

Below the table, there is a checkbox for 'Start after created' which is unchecked. At the bottom, there is an 'Advanced' checkbox (checked), and 'Back' and 'Finish' buttons.

Figura 9.4: Formulari de creació de màquines virtuals (2)

9.3.2 Sistema operatiu

A la fundació Guifi.net pràcticament només es treballa amb servidors Linux³ per la seva condició de software lliure i també per la robustesa que els caracteritza.

Una distribució de Linux és el conjunt d'eines de sistema i aplicacions que juntament amb el kernel Linux són capaces de dotar de funcionalitat una màquina. Per a realitzar aquest projecte s'ha fet servir l'última versió de la coneguda distribució Linux **Debian**. Podríem definir la distribució **Debian** amb els següents punts:

- Fa molt èmfasi en el codi lliure. Al seu repositori principal només hi ha software que compleix amb aquestes característiques. El software que no és lliure es troba al repositori *non-free* que no està inclòs per defecte.
- És el sistema operatiu universal. Això vol dir que s'executa en quasi qualsevol plataforma: Des d'un processador ARM a un Mainframe IBM z, passant per un PC x64.
- Conté versions molt estables del software als seus repositoris, motiu pel qual és un molt bon sistema operatiu en servidors.



Figura 9.5: Logotip de **Debian**

9.3.3 Software instal·lat i configuracions

Les màquines virtuals basades en **Debian stretch 9** compten amb el següent software:

- Docker 18.09 ce
- node.js 8.15.0

³Cal recordar que Linux només és el kernel, el conjunt del sistema operatiu basat en Linux s'anomena distribució.

- npm 6.4.1
- apache 2.4.25
- mariaDB 10.1.37
- certbot (Let's encrypt SSL certificats)

La majoria de paquets s'han instal·lat des dels repositoris oficials de la distribució no obstant els paquets de Docker i node.js s'han instal·lat des dels repositoris dels mantenidors.

Tots els serveis exposats públicament utilitzen apache com a reverse proxy per tal de mantenir els sistemes segurs. A més tant el *backend* com el *frontend* compten amb certificats SSL reconeguts. Per tal de poder adquirir-los s'ha utilitzat el servei gratuït Let's Encrypt (<https://letsencrypt.org/>).

El *backend* s'executa al sistema en forma de dimoni⁴ gràcies a un servei de systemd. S'ha creat una unitat de systemd força simple que es pot trobar a l'apèndix G.1.

Finalment també s'ha instal·lat una instància de MapServer amb els scripts de guifimaps. Es va valorar que l'opció més simple per posar-ho en marxa era executar directament el software dins d'un contenidor **Docker**. Aprofitant la imatge ja creada per als mapes (veure 8.1.2) s'ha executat com a dimoni. Al següent fragment mostra la comanda utilitzada per a córrer el contenidor **Docker** a producció:

```
$ docker run --name guifimaps -d -p 127.0.0.1:9090:80 --restart always -e "
GUIFI_WEB=guifi.net" -e "FIBERFY_API=fiberfy-server:1337" -e "
FIBERFY_LOGIN=/auth/login" -e "FIBERFY_EXPORT=/api/v1/export/all" -e "
FIBERFY_USERNAME=test" -e "FIBERFY_PASSWORD=<password>" --add-host
fiberfy-server:10.90.230.43 -v "$PWD"/guifimaps:/home/guifi/guifi/
guifimaps
```

9.4 Monitoratge

Un aspecte molt important de qualsevol sistema en producció és la disponibilitat. Per garantir que un servei sempre estigui en marxa s'ha de realitzar un control d'aquest i en cas de fallada s'ha de restaurar el servei. Els requisits de monitoratge d'aquest projecte són els següents:

- Seguiment dels processos: **apache2**, **fiberfy-server**.
- Seguiment del container: **guifimaps**.
- Intentar reiniciar els serveis afectats en cas d'aturada.
- Notificar fallades (tant si s'han restaurat els serveis com sinó)

⁴A producció de moment no es fa servir Docker per exposar el *backend*

Al mercat hi ha diferents alternatives per monitorar sistemes. Una de les alternatives més conegudes és nagios.

No obstant s'ha descartat aquesta eina degut a la seva complexitat, ja que per monitorar una màquina virtual no es necessita una plataforma tant potent. Per dur a terme aquesta tasca s'ha considerat l'ús d'una petita eina de monitoratge molt simple i lleugera: **monit**. Aquesta eina està programada en C i funciona sobre qualsevol entorn *UNIX*. Permet entre altres coses:

- Seguiment de processos i serveis.
- Seguiment de l'ús de recursos del sistema.
- Presenta una interfície web simple i una **API REST**.
- Permet actuar en cas de fallades de forma automàtica.
- Permet notificar alertes.



Figura 9.6: Logotip de **monit**

9.4.1 Instal·lació i configuració

Per tal de posar en marxa el software **monit** hem d'instal·lar el paquet que el conté. Tal i com hem comentat a la secció 9.3 totes les instàncies tenen com a sistema operatiu la distribució Linux *Debian stretch* que utilitza el sistema de paquets **apt**:

```

1 $ sudo apt install monit
2 $ sudo systemctl status monit # Aquí ens apareixerà l'estatus del servei (ha
    d'estar actiu)

```

Per defecte el servidor web de gestió de l'eina es troba desactivada, per utilitzar-la s'ha modificat el fitxer `/etc/monit/monitrc` amb el següent contingut:

```

1 set httpd port 2812 and
2     use address localhost
3     allow localhost
4     allow admin:<password> # Afegim usuari i contrasenya

```

Amb aquesta configuració ja ens apareixerà un petit formulari amb informació bàsica del sistema. A la figura 9.7 podem veure'n una captura.

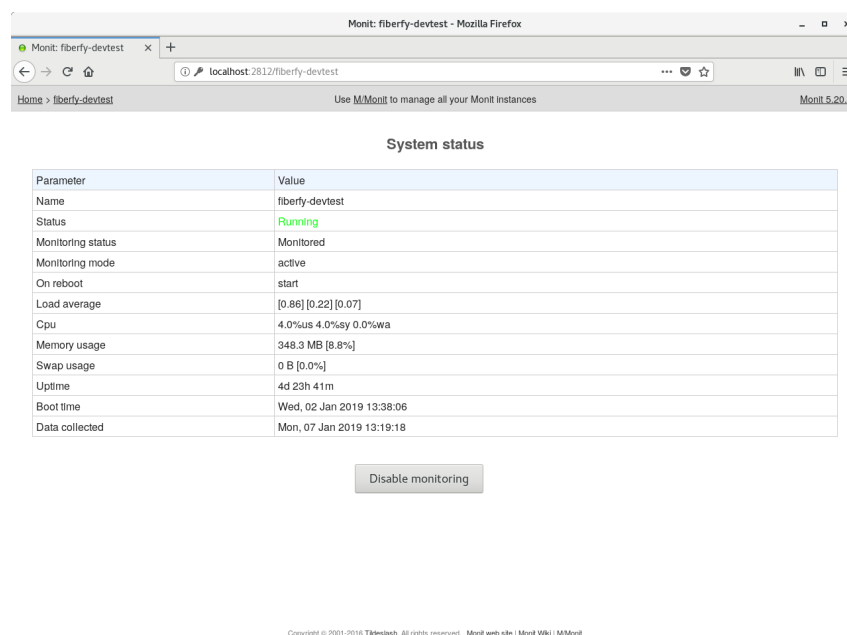


Figura 9.7: Captura inicial de **monit**

El propòsit principal per utilitzar aquesta eina és la possibilitat de monitorar serveis en execució. Per tal de fer un seguiment d'un procés s'ha d'especificar el comportament en un fitxer de configuració. S'ha creat aquest fitxer a l'ubicació `/etc/monit/conf.d/fiberfy-server` amb el contingut següent:

```

1 check process fiberfy-server
2     matching "/usr/bin/node_app.js"
3     start program = "/bin/systemctl start fiberfy-server" with timeout
        5 seconds

```

```

4      stop program = "/bin/systemctl stop fiberfy-server" with timeout 5
        seconds
5      if not exist then restart
6      if not exist then alert
7      if 3 restarts within 8 cycles then timeout
8      if does not exist then alert

```

D'aquesta manera quan el servei cau, **monit** intentarà reiniciar el servei de forma automàtica i a més avisarà d'aquesta fallada. Si el problema persisteix al reiniciar 3 vegades el servei s'envia un *timeout* que implica deixar d'actuar i avisar de la situació. Per defecte l'aplicació no avisa més enllà d'utilitzar fitxers de log dins del mateix sistema. Un requisit és comunicar a l'equip de DevOps la fallada del sistema perquè pugui actuar, per tant s'ha decidit configurar un compte de correu electrònic per notificar els avisos. La configuració de la missatgeria es fa des del mateix fitxer de configuració *monitrc*. El següent fragment ens mostra com activar el servei de correu:

```

1 set mailserver authsmtp.rgfcomputing.com port 587
2     username "smtp@rgfcomputing.com" password <password>
3     with timeout 30 seconds
4
5 set mail-format \{ from: monit@rgfcomputing.com \}
6
7 set alert roger.garcia@guifi.net # receive all alerts

```

9.5 Networking (testing)

9.5.1 Carrier service

El servidor de producció es troba al CPD de Guifi.net a Barcelona. No obstant el servidor de testing es troba a La Garriga. A diferència de les línies residencials que s'ofereixen al mercat minorista, aquest servidor es troba connectat a un circuit de fibra (FTTH GPON) fins al CPD de Barcelona. D'aquesta manera se li poden assignar tantes direccions IP Públiques com sigui necessari a més de poder enrutar prefixes de Guifi.net (IPs 10.x.x.x) sense l'overhead de realitzar túnels a través d'Internet.

Realment la tecnologia que es fa servir és la mateixa que es pot trobar a qualsevol fibra residencial amb la diferència que en aquest cas l'operadora propietària de la infraestructura d'accés enlloc de realitzar l'enrutament a Internet, entrega el trànsit a nivell 2 de la capa OSI[12]⁵ al backbone de Guifi.net.

⁵OSI és una especificació d'interconnexió de xarxes que serveix per descriure-les. En el cas del nivell 2 estem parlant d'el nivell d'enllaç; per a connexions L2 ethernet seria com si hi hagués 2 ports de xarxa, un a cada punta.

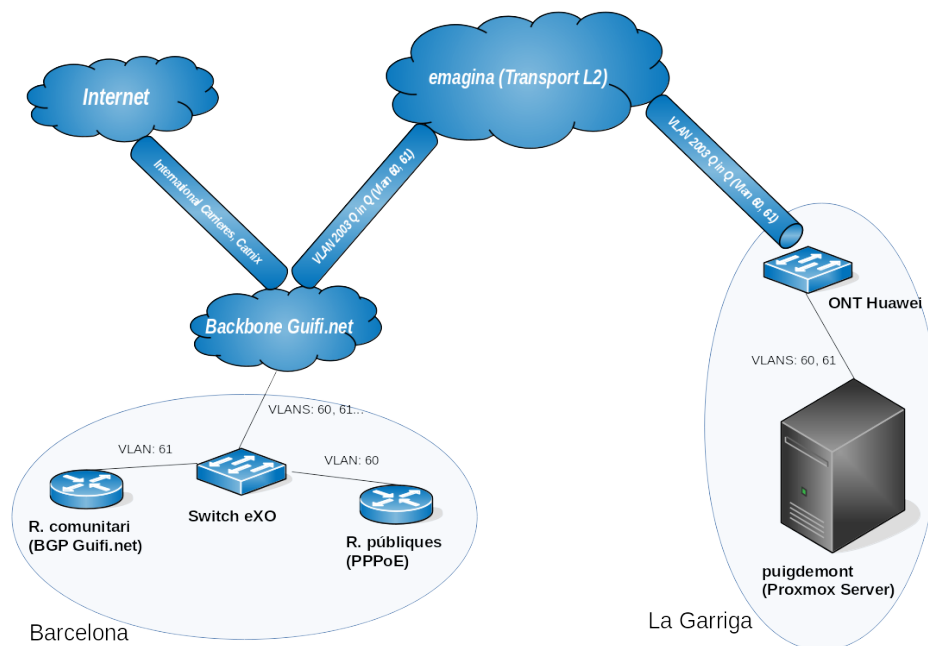


Figura 9.8: Diagrama sobre la topologia de xarxa (L2)

9.5.2 Xarxa Local

Dins del servidor arriben dues VLANs la 60 i la 61. La VLAN 60 serveix per establir les sessions PPPoE que són necessàries per obtenir les direccions IP públiques subministrades.

Internament el servidor *puigdemont*, que té instal·lada una plataforma de virtualització basada en Debian, utilitza el bridge del kernel de Linux per tal de connectar les diferents interfícies virtuals a les interfícies físiques. A la taula 9.9 es poden veure totes les interfícies que presenta el host.

| Node 'puigdemont' | | | | | | | | | | |
|-------------------|------------|----------------|--------|-----------|-----------|--------------|--------------|-----------------|-------------|------------------------|
| | Create | Revert | Edit | Remove | | | | | | |
| | Name | Type | Active | Autostart | VLAN a... | Ports/Slaves | IP address | Subnet mask | Gateway | Comment |
| | enp0s25 | Network Device | Yes | No | No | | | | | |
| | enp0s25.10 | Linux VLAN | Yes | Yes | No | | | | | |
| | enp0s25.60 | Linux VLAN | Yes | Yes | No | | | | | |
| | enp0s25.61 | Linux VLAN | Yes | Yes | No | | | | | |
| | vmbr0 | Linux Bridge | Yes | Yes | No | enp0s25 | 192.168.4.10 | 255.255.255.0 | 192.168.4.1 | Local network (RGFNET) |
| | vmbr1 | Linux Bridge | Yes | Yes | No | enp0s25.10 | 10.90.230.34 | 255.255.255.224 | | Guifi Network |
| | vmbr2 | Linux Bridge | Yes | Yes | No | enp0s25.60 | | | | Public IPs eXO |
| | vmbr3 | Linux Bridge | Yes | Yes | No | enp0s25.61 | | | | BGP links |

Figura 9.9: Taula amb totes les interfícies del host *puigdemont*

Capítol 10

Gestió econòmica

Tots els costos analitzats deriven de les tasques esmentades al gràfic de *Gantt*. S'analitzen els costos de personal, de hardware, de software i altres costos.

10.1 Recursos humans

Aquest projecte ha estat realitzat per una única persona amb l'excepció de col·laboracions esporàdiques d'un altre desenvolupador amb més experiència (unes 4 hores setmanals). Per estimar els costos d'aquesta persona ens basarem en el cost per hora del desenvolupador principal i responsable del projecte i li aplicarem un coeficient de 1.5. A la realitat aquesta persona no representa un cost sobre el projecte ja que les hores que hi dedica són comptabilitzades com a voluntàries dins d'un conveni de cooperació, però si comptabilitzem els costos reals segurament l'estimació donada estaria més a prop de la realitat. Pel que fa l'equip directiu considerarem que el seu cost hora és equivalent al segon desenvolupador.

| Fase | Rols | | | Cost estimat |
|---|----------------|---------------|--------------|----------------|
| | Equip directiu | Dev principal | Dev auxiliar | |
| Definició del projecte | 10h | 30h | 10h | 930 € |
| Gestió del projecte | 0h | 80h | 0h | 1240 € |
| Preparació d'entorns de desenvolupament | 0h | 30h | 0h | 465 € |
| Actuacions sobre el <i>backend</i> | 10h | 230h | 20h | 4262,5 € |
| Actuacions sobre el <i>frontend</i> | 10h | 240h | 20h | 4417 € |
| Actuacions sobre el model de dades | 10h | 80h | 0h | 1472,5 € |
| Integració dels mapes | 0h | 60h | 0h | 930 € |
| Posada a producció | 0h | 40h | 0h | 620 € |
| Documentació | 0h | 120h | 10h | 2092,5 € |
| Total | 40h | 910h | 60h | 16429 € |

Partim de la base que coneixem el cost/hora del desenvolupador principal. Aplicant un factor de 1.5 contra als altres 2 perfils obtenim la següent taula:

| | Rols | | |
|------|----------------|---------------|--------------|
| | Equip directiu | Dev principal | Dev auxiliar |
| Cost | 23,25€/hora | 15,5 €/hora | 23,25 €/hora |

10.2 Recursos *hardware*

| Producte | Preu u. | Vida útil | Amortització/h | Hores ús | Amortització total |
|---------------------------------------|---------|-----------|----------------|----------|--------------------|
| Portàtil PCSpecialist Core i5 7th Gen | 990 € | 3 anys | 0,1916 €/hora | 910 | 174,39 € |
| Màquina virtual Proxmox | 100 € | 1 any | 0,0114 €/hora | 2880 | 32,87 € |

Per fer les estimacions d'ús del equip portàtil tenim en compte que a l'any es treballen 1722 hores. Aleshores si tenim en compte que un portàtil d'aquestes característiques s'amortitza a tres anys, tenim un total de 5166 hores per amortitzar. En el cas de la màquina virtual hauriem de comptar les 24 hores del dia els 365 dies de l'any (sempre encesa).

10.3 Recursos de *software*

Nosaltres com a entitat compromesa amb el *software* lliure, pràcticament només fem servir *software* amb aquest tipus de llicències. En el cas d'aquest projecte podem dir que està fet íntegrament amb *software* sense llicències propietàries, motiu pel qual no hem de pagar a ningú per usar-lo.

10.4 Altres

Després d'analitzar els costos de personal, *hardware* i *software* ens trobem uns altres costos:

- **Costos indirectes:** Aquests fan referència als costos d'oficina. En el nostre cas, l'ajuntament de Gurb, ens cedeix un espai municipal per a la realització de les nostres tasques ja que aportem molt valor social al municipi.

- **Contingència:** La contingència cobrirà els sobre costos en cas de que hi hagi alguna desviació en els terminis o algun imprevist. Aquests imprevistos venen descrits al resum de tasques i al gràfic de *Gantt*. Nosaltres tenim en compte el tipus de projecte hem triat una contingència del **10%** sobre el total dels costos. Això ens dona un total de **1663,62 €**.
- **Imprevistos:** Com a imprevist podríem tenir que la màquina virtual fallés per culpa d'una fallada dels nostres sistemes. En cas de que això succeís i volguéssim seguir oferint servei, hauríem de llogar un **VPS** en algun proveïdor extern com **Amazon**. Això tindria un cost d'uns 10 € al mes aproximadament. Per tant, tenim en compte el baix import, reservem 10 € per aquesta possible incidència.

10.5 Resum de costos

| Personal | Hardware | Software | Indirectes | Contingència | Imprevistos | Total |
|----------|----------|----------|------------|--------------|-------------|------------|
| 16429 € | 207,26 € | 0 € | 0 € | 1663,62 € | 10 € | 18309,88 € |

10.6 Control de gestió

El control de gestió s'ha fet mitjançant reunions mensuals entre l'equip de desenvolupadors i l'equip directiu. Aquestes reunions han tingut una durada aproximada d'1 hora i han servit per analitzar el progrés de les diferents tasques i les incidències que ens anàvem trobant. Les fases de desenvolupament més crítiques han estat les dues implementacions (*backend* i *frontend*). Tot-hi així, al tractar-se d'un projecte de desenvolupament de software el cost més important i també la font principal de desviacions són els recursos humans.

Inicialment el projecte s'havia projectat per tenir una durada aproximada de 1000 hores (a la descripció de tasques dels recursos humans ens dona un total de 1010 hores). A la realitat el projecte ha acabat durant 1100 hores. Això dona una desviació temporal d'un **8,91 %**. Si calculem l'increment de cost tenim:

- L'increment d'hores ha estat íntegrament sobre les tasques fetes pel desenvolupador principal (Qui té la major participació en el projecte)
- Tenim el cost/hora que és de **15,5 €**
- L'increment és de **90 hores**

L'increment de cost seria: $I = 90h \times 15,5e/h = 1395e$ El percentatge de la desviació respecte l'import total del projecte (sense contingència) és de: $P = (1395e / (18309,88e - 1663,62e)) \times 100 = 8,38\%$

Aquest import representa menys d'un 10 % del total del projecte i per tant es pot assumir amb la contingència assignada.

Capítol 11

Sostenibilitat

11.1 Ambiental

Actualment, els participants del projecte de Guifi.net (bàsicament petits operadors locals i associacions d'usuaris), fan servir eines pròpies o de tercers per tal de documentar les xarxes de fibra òptica que despleguen. El que es proposa en aquest projecte és desenvolupar una solució que pugui servir per tots i que estigui *hostejada* en un únic servidor comú per tots. D'aquesta manera es poden reduir les emissions de CO₂ i altres gasos nocius que s'emeten al generar corrent elèctric. Un altre aspecte rellevant d'aquest servidor és que no està només dedicat al projecte documental de fibra sinó que també serveix per tenir molts altres serveis que s'ofereixen en el context de guifi.net (web del projecte, web de la fundació, **NOC**...), és a dir que es reutilitza un mateix recurs per a diferents propòsits. D'altra banda també cal remarcar que en el desenvolupament del projecte només s'ha fet servir un ordinador portàtil que té unes emissions de gasos d'efecte hivernacle en el seu ús negligibles (no en la seva fabricació).

11.2 Econòmica

Per tal de realitzar aquest projecte s'han analitzat els recursos dels que disposavem (hardware) i també de totes les alternatives lliures que hi havia que ens poguessin resoldre el nostre problema. Com que no hi ha alternatives lliures que ens serveixin pels nostres propòsits n'hem desenvolupat una; hem utilitzat eines i llibreries publicades amb llicències lliures per tal d'estalviar el màxim de recursos.

El projecte aportarà una millora econòmica evident al projecte de Guifi.net. Primer de tot, es compartirà una única plataforma documental entre el conjunt d'entitats que formen la xarxa de Guifi.net. Això permetrà repartir els costos d'un sistema entre molts usuaris i no un sistema per a cada usuari. D'altra banda a mig termini també ens permetrà reduir costos al tractar-se d'una solució desenvolupada per nosaltres i oberta, que no ens generarà un cost recurrent d'ús de forma sostinguda en el temps.

11.3 Social

Des del meu punt de vista el fet d'haver realitzat aquest projecte m'ha reportat uns coneixements tecnològics i socials immensos. Tecnològicament parlant, he après com funcionen tecnologies molt diverses i punteres en la indústria informàtica actual. De la vessant social he pogut aprendre a tractar amb un entorn real de problemes sense resoldre i ser capaç de proposar una solució que cobrés aquesta necessitat. A més al tractar-se d'un projecte basat en els comuns encara considero que ha estat més interessant.

Hi havia una necessitat clara de desenvolupar aquesta aplicació ja que la motivació a l'hora de plantejar el projecte va venir donada per les exigències que ens feien les empreses que instal·laven fibra dins del projecte de Guifi.net. Aleshores al ser una proposta que ve del món de l'empresa i les organitzacions està clar que té una certa demanda dins del nostre projecte.

L'impacte social que pot tenir aquest projecte té una afectació indirecte. Al reduir els costos de mantenir un sistema documental de fibra (al compartir-lo entre moltes organitzacions) permet reduir costos de forma global i això ens permetrà invertir aquests diners estalviats en promoure més desplegaments de fibra de comuns a zones del territori on encara no disposen d'aquesta tecnologia. L'impacte que té arribar amb fibra a les zones rurals és molt gran i positiu per les persones i empreses que hi viuen.

En l'anàlisi fet no s'observa que degut a la implementació d'aquest problema es redueixin llocs de treball ja que no era una feina que repercutís directament a un tipus de professió, més aviat és una eina que ajudarà a ser més productius als treballadors d'organitzacions que actualment tiren endavant projectes de fibra (enginyers en telecomunicacions, tècnics de xarxa o instal·ladors).

Capítol 12

Conclusions

12.1 Resultats

Gràcies a la realització d'aquest projecte s'ha avançat molt a l'hora de gestionar els desplegaments existents de fibra òptica dins de Guifi.net. S'han complert els objectius inicials plantejats per a la realització del projecte, motiu pel qual el projecte ja es troba a producció complint part de la seva funcionalitat.

Mitjançant l'ús de tests funcionals de l'API el projecte s'ha pogut posar a producció de forma senzilla sense reportar un gran nombre de problemes. L'ús d'entorns de desenvolupament personalitzats també han estat d'ajuda per permetre muntar les integracions dels diferents sistemes de forma còmode i eficient.

Des d'un punt de vista ja més personal, el fet de poder realitzar un treball molt enfocat a la documentació i inventari de xarxes de fibra òptica m'ha permès conèixer amb força detall com funcionen aquest tipus d'infraestructures. Un altre aspecte interessant d'haver estat treballant a Guifi.net és el d'adquirir uns coneixements més amplis en xarxes de computadors a nivell d'operador fent un ampli ús d'estàndards que no es troben a l'abast del públic general. Finalment un altre aspecte molt interessant de realitzar aquest treball ha estat l'ampli coneixement adquirit a l'hora de treballar amb el desenvolupament d'aplicacions progressives i basades en SPA.

12.2 Millores de futur

Tot-hi que el projecte realitzat és força complet i soluciona gran part dels problemes que hi havia l'hora de documentar els desplegaments de fibra òptica encara hi ha algunes limitacions a millorar:

- Actualment només es poden afegir elements a dins de projectes amb una relació 1 a N (on N fa referència a l'element i 1 el projecte), això implica que sigui impossible compartir elements entre els projectes. Per solucionar aquesta limitació s'hauria de modificar el model de dades i a més canviar de forma substancial el frontend.

- No hi ha una estructura ACL massa ben definida. Actualment només hi ha un tipus d'usuari que poden veure el 100 % de la informació i només editar els seus projectes i donar permisos d'escriptura a altres usuaris. També s'hauria d'incloure ACLs per gestionar els tipus d'entitats del projecte (no tothom ho hauria de poder consultar tot).
- Els mapes són de tipus raster. Això vol dir que estan formats per imatges quadrades (tiles) ja compostes. A dia d'avui amb la potencia dels ordinadors actuals és molt més interessant l'ús de mapes vectorials renderitzats pel propi client ja que acostumen a oferir una experiència d'usuari més bona. Amb tecnologies com WebGL no és molt difícil d'implementar.
- S'hauria de millorar l'usabilitat del frontend per fer-lo més eficient a l'hora d'importar grans quantitats d'informació: Això inclou permetre aplicar massivament propietats a les diferents entitats o per exemple crear punts a l'interior de trams de fibra ja creats.

Appendix

A Manual fiberfy-nt

Manual d'usuari *fiberfy-nt*



11 d'octubre de 2018

| Historial de versions | | |
|-----------------------|------------|---|
| Versió | Data | Canvis |
| prototip | 10/08/2017 | Primera edició del document. Versió antiga del fiberfy |
| 18.10 | 11/10/2018 | Primera versió estable de l'aplicació. Canvis rellevants: <ul style="list-style-type: none"> • El frontend canvia el nom a fiberfy-nt • Es refà per complet l'aplicació només mantenint els mapes • S'incorpora un importador de projectes • S'incorpora un nou tipus de localització: No Definit • Es demana confirmació al eliminar entitats • S'incorpora un buscador de projectes |

Índex

| | | |
|----------|--|-----------|
| 1 | Introducció | 4 |
| 2 | Accés al portal | 4 |
| 3 | Funcionament del mapa | 4 |
| 4 | Gestió de projectes | 5 |
| 4.1 | Edició | 6 |
| 5 | Obra civil | 8 |
| 5.1 | Importador | 8 |
| 5.2 | Exportador | 8 |
| 5.3 | Gestió de llocs | 9 |
| 5.3.1 | Creació | 9 |
| 5.3.2 | Edició | 11 |
| 5.4 | Gestió de <i>trams</i> | 12 |
| 5.4.1 | Creació | 12 |
| 5.4.2 | Edició | 12 |
| 6 | Cablejat | 14 |
| 6.1 | Gestió de caixes | 14 |
| 6.1.1 | Creació | 14 |
| 6.1.2 | Edició | 14 |
| 6.2 | Gestió de fibres/tubs/cables | 15 |
| 6.2.1 | Creació | 15 |
| 6.2.2 | Edició | 15 |
| 6.3 | Gestió de fusions | 17 |
| 6.3.1 | Creació i edició | 17 |
| 6.3.2 | Visualització | 18 |

1 Introducció

L'eina de desplegaments de fibra òptica *fiberfy-nt* ha estat creada per facilitar la confecció de projectes als diferents operadors de la xarxa **Guifi.net**. Aquest document és manual d'usuari per ensenyar les característiques de l'aplicació i explicar-ne el seu funcionament.

2 Accés al portal

L'aplicació *fiberfy-nt* és un servei web accessible des de qualsevol dispositiu amb un navegador compatible amb **JavaScript**. La seva direcció és: <https://fiberfy.guifi.net/>

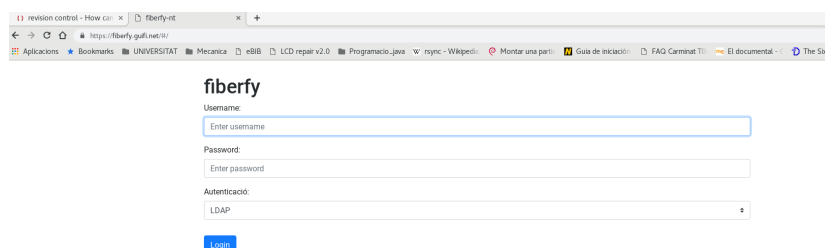


Figura 1: Pantalla de login *fiberfy-nt*

Els paràmetres d'accés vindran subministrats per la fundació. Actualment es pot autenticar usuaris mitjançant l'**LDAP** del **GLIR**, per tant l'usuari podrà usar el mateix usuari pels diferents serveis que s'ofereixen. Un cop l'usuari s'ha autenticat li apareixerà un mapa de Catalunya.

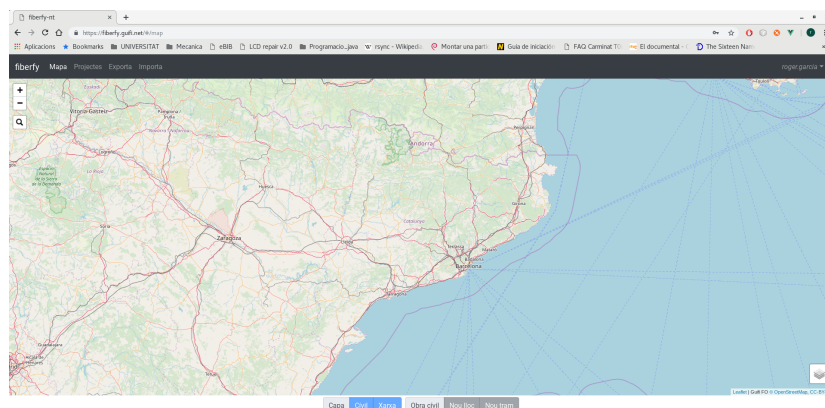


Figura 2: Mapa d'inici *fiberfy-nt*

3 Funcionament del mapa

El mapa ens permet treballar amb diferents capes. Hi ha dos tipus de capes:

- Capes base: Aquestes són les capes bàsiques que ens gràfiquen l'esquelet del mapa; només s'en pot seleccionar una a la vegada. A dia d'avui tenim la capa gratuïta d'**OpenStreetMap** i diverses capes de **Google**.

- Capes auxiliars: Aquestes són capes que es mostren a sobre de la capa base i ens aporten informació extra. Actualment disposem de 4 capes de Guifi.net (nodes, enllaços, localitzacions (fibra), *trams* (fibra)).

Per interactuar amb les diferents capes del mapa situarem el punter sobre un pictograma situat al extrem inferior dret com podem veure a la figura 3:



Figura 3: Pictograma gestió de capes

A la figura 4 podem veure el gestor de capes del mapa.

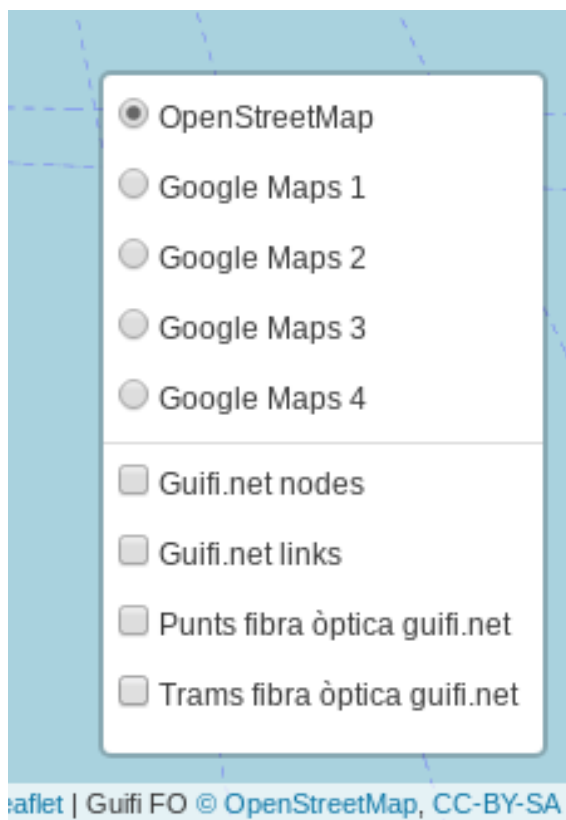


Figura 4: Menú de gestió de capes

4 Gestió de projectes

Per tal de crear projectes haurem d'accedir al menú situat a la capçalera de la pàgina i accedirem a la pestanya de projectes.

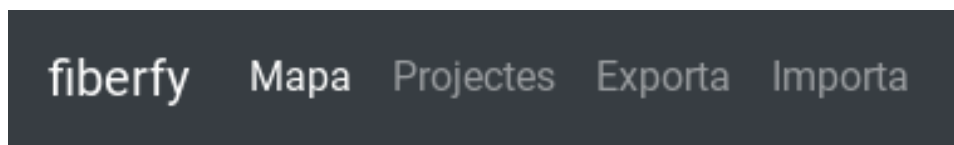


Figura 5: Menú superior *fiberfy-nt*

Al formulari de projectes ens apareixerà primer una petita targeta amb la informació del projecte seleccionat que ens permetrà modificar les seves característiques:

- *Guardar Posició*: Aquest botó ens guarda la posició del mapa actual al projecte. Així quan tornem a utilitzar l'aplicació ens mantindrà aquesta posició i no hauré de navegar pel mapa buscant-lo.
- *Editar*: Aquest botó ens porta a un formulari on podrem modificar els atributs del projecte

Ens apareixerà un buscador de projectes i a la part inferior es mostraran els resultats obtinguts. Per defecte es mostren tots els projectes; si l'usuari desitja mostrar tots els projectes després d'haver fet una búsqueda haurà d'esborrar el text i tornar a prémer **Buscar**. Els projectes de cada usuari seran visibles per tots els usuaris i només podran ser modificats per ell mateix. Per tal de treballar sobre qualsevol projecte aquest s'haurà d'activar prement el botó *Activar* situat a la barra de botons de la dreta.

Projectes

Projecte actual

stats

ID: 11

Status: Projectat

Zona: 2436

Desactivar Guardar pos. Editar Eliminar

Llistat de projectes Afegir projecte

Nom: Buscar

| | | | |
|------------------|------------|--------------|-----------------|
| Test sdjksdhjksd | Activar | Guardar pos. | Eliminar |
| Alt Urgell Nou | Activar | Guardar pos. | Eliminar |
| stats | Desactivar | Guardar pos. | Editar Eliminar |
| New test 2 | Activar | Guardar pos. | Eliminar |

Figura 6: Formulari de projectes *fiberfy-nt*

Per tal de crear nous projectes haurem de prémer el botó verd **Nou projecte** situat a la part superior dreta del Llistat de projectes.

4.1 Edició

Per tal d'editar projectes l'usuari haurà de prémer el botó blau *Editar* i serà conduït a un formulari com el que podem veure a la següent figura:

Editar projecte

Nom:

Localització:

Zoom:

Status:

Zona:

Actualitzar

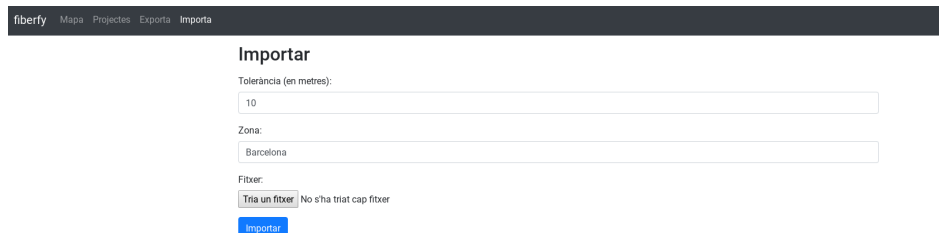
Figura 7: Formulari de projectes *fiberfy-nt*

Aquí l'usuari podrà modificar aspectes com el nom del projecte, el punt de referència del mapa (Localització + zoom), status (el mateix que hi ha a la web de Guifi.net) i finalment la zona per defecte. La zona ve donada per la base de dades de Guifi.net (la mateixa que la web), l'usuari la pot seleccionar d'un desplegable a mesura que va escrivint.

5 Obra civil

5.1 Importador

Una de les novetats més importants del *fiberfy-nt* és la possibilitat d'importar mapes existents de punts i línies i tractar-los com si de localitzacions i trams es tractessin. Per tal d'importar informació necessitem haver creat prèviament un projecte i tenir-lo activat. Un cop complim aquest requisit podem accedir a la pestanya d'*Importa* i podrem veure un formulari com el següent:



The screenshot shows the 'Importar' (Import) form in the fiberfy application. The form is located under the 'Importa' tab in the top navigation bar. It contains the following fields and elements:

- Tolerància (en metres):** A text input field with the value '10'.
- Zona:** A text input field with the value 'Barcelona'.
- Fitxer:** A section with a button labeled 'Tria un fitxer' and a text label 'No s'ha triat cap fitxer'.
- Importar:** A blue button at the bottom of the form.

Figura 8: Formulari d'importació d'obra civil

La tolerància és el marge que té l'algoritme per unir trams amb llocs. Els estàndards de **GIS** no contemplen la possibilitat d'acoplar punts i rectes, per tant a l'hora d'establir els enllaços l'aplicació els ha de predir. El marge de tolerància indica en metres la distància màxima d'unió de rectes en un únic punt. La zona és la mateixa que s'introduiria al crear un node de Guifi (per defecte la del projecte). Finalment hem de seleccionar el fitxer que volem importar. Els formats suportats són:

- **KML**
- **KMZ**
- **GeoJSON**

En cas de disposar d'altres formats es poden convertir a qualsevol dels compatibles. Igualment sempre hi ha la possibilitat d'obrir un *ticket* per tal de resoldre qualsevol incidència.

5.2 Exportador

Aquesta funcionalitat ens permet exportar tota l'obra civil d'un projecte en el format **GeoJSON**. Per tal de fer-ho servir necessitem haver seleccionat un projecte prèviament i accedir a la pestanya *Exporta*. Aleshores ens apareixerà un formulari amb un botó per descarregar el fitxer:

Exporta

Descargar

Figura 9: Formulari d'exportació d'obra civil

5.3 Gestió de llocs

Els llocs dins de l'aplicació simbolitzen els punts físics d'obra civil on s'ubicaran les caixes i sistemes de telecomunicacions. Contenen unes coordenades i atribut físic que ens indica el tipus d'element. També permeten afegir comentaris i modificar-ne el nom.

5.3.1 Creació

Sobre el mapa inicial podem crear diferents llocs on s'ubicaran els equips de fibra òptica. Per tal de crear nous llocs hem de seleccionar el botó *Nou lloc* situat a la barra inferior dreta¹. Aleshores hem d'anar marcant amb el botó esquerre del ratolí els punts on hi volem col·locar un nou *lloc*. Per sortir del menú de creació de nous llocs podem clicar un altre cop sobre el botó *Nou lloc* o prémer la fletxa enrere verda.

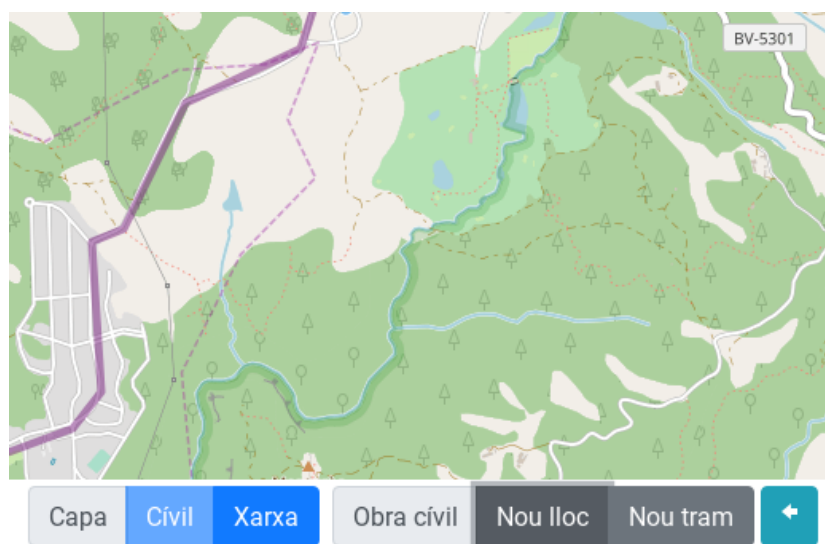


Figura 10: Menú de creació llocs

Els llocs creats se simbolitzaran amb els següents marcadors:

¹És necessari seleccionar la capa d'Obra Cívil

| Símbol | Explicació |
|---|---|
|  | Símbol no definit, quan no coneixem el tipus. |
|  | Arqueta situada al sòl. |
|  | Poste |
|  | Ganxo |
|  | Poe |
|  | Armari |
|  | Cambra |
|  | Salt |

Per defecte tots els llocs creats seran de tipus **No Definit**.

5.4 Gestió de *trams*

Els *trams* dins de l'aplicació ens indicaran l'obra civil encarregada d'unir uns determinats llocs. Podrem definir com serà aquest tram i escriure alguna anotació(observació).

5.4.1 Creació

Per tal de crear *trams* haurem de seleccionar el botó *Crea Tram* del menú d'Obra Civil². Un cop seleccionat el botó hem de clicar sobre el primer *lloc* que volem unir i anar clicant sobre els següents llocs adjacents tot traçant una línia.

En cas de voler desfer la creació del tram només hem de prémer el botó verd amb la fletxa enrera.

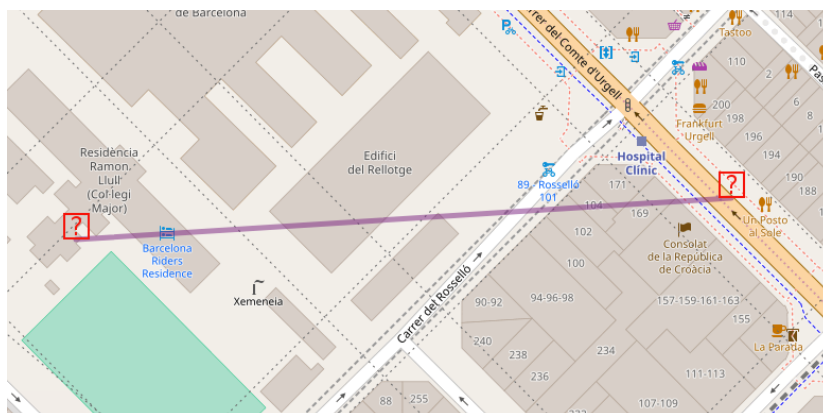


Figura 13: Mapa amb un tram

Si volem unir diferents llocs amb un recorregut no recte, podem afegir punts al traçat marcant els punts de vèrtex sobre el mapa després d'haver marcat el primer *lloc* i tancant el traçat al marcar el segon *lloc*

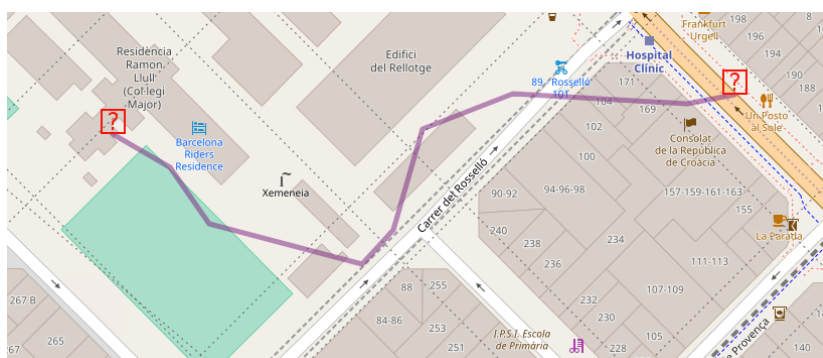


Figura 14: Mapa amb *trams* no rectes

Per sortir del menú de creació de nous *trams* podem clicar a qualsevol tram existent.

5.4.2 Edició

Per editar un simplement haurem de col·locar el punter a sobre i clicar. Aleshores ens apareixerà un formulari amb tots els atributs que podem alterar.

²És necessari seleccionar la capa d'Obra Cívica

Editar tram: 1312

Nom:

Primer lloc:

Segon lloc:

Distància:

Tipus:

Status:

Observacions:

Figura 15: Formulari d'edició de *trams*

Aquí podem modificar tots els atributs del tram. En cas de voler alterar el *lloc* d'inici i *lloc* final ens apareixerà un auto-completat que ens permetrà seleccionar entitats de dins del projecte; tot-hi així no se'n recomana el seu ús per evitar possibles desajustos en els camins.

Si hem fet alguna modificació i la volem guardar haurem de clicar el botó *Actualitzar*. Si volem eliminar el tram haurem de clicar el botó *Eliminar* i automàticament serem redirigits al mapa.

6 Cablejat

6.1 Gestió de caixes

Les caixes són els elements de telecomunicacions per tal de connectar i distribuir les diferents fibres. Aquests elements són necessaris al inici i final d'una fibra.

6.1.1 Creació

Per tal de crear *caixes* necessitarem haver definit prèviament llocs (d'obra civil). Un cop complert aquest requisit seleccionarem la capa *Xarxa*. Després ja podem clicar qualsevol dels llocs creats anteriorment. Si no contenen *caixes* apareixeran pintats en gris; en cas contrari estaran pintats en taronja. Aleshores s'ens mostrarà el següent formulari:

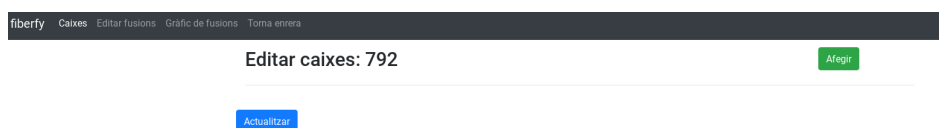


Figura 16: Menú de *caixa* buit

Per tal de seguir amb el procés haurem de prémer el botó verd *Afegir*. Aleshores ja tindrem una *caixa* creada amb la següent aparença:

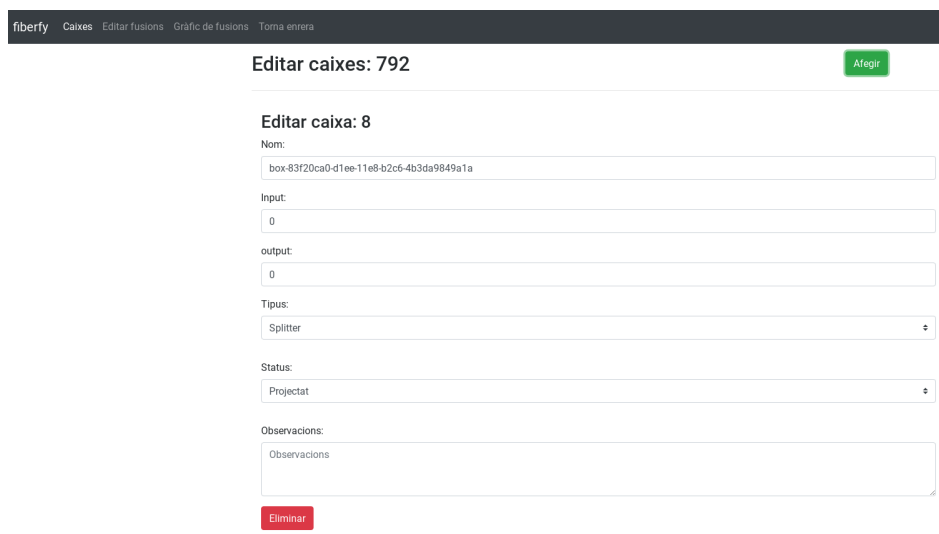


Figura 17: Menú de *caixa* omplert

6.1.2 Edició

Per tal de modificar els atributs d'una *caixa* existent només caldrà clicar sobre la capsula en qüestió des de la capa de Xarxa de l'eina. Aleshores ens apareixerà el formulari de creació de *caixes* amb les caixes existents en aquell *lloc*. Per tal de modificar els atributs haurem de validar els canvis utilitzant el botó *Actualitzar* del formulari. Per eliminar caixes haurem de prémer el botó *Eliminar*.

Dins d'aquest formulari també podrem establir el tipus de *caixa* que és i el nombre de fibres d'entrada i sortida que passaran per elles. Aquestes dades seran importants a l'hora de realitzar les fusions de fibres.



Figura 19: *Popup selecció de cables*

Un cop seleccionat el *cable* desitjat s'ens mostrarà un formulari com el següent:

fiberty
Toma entera

Editar cable: 8

Nom:

Status:

Observacions:

Template:

Editar tubs: 8

Afegir

Actualizar

Eliminar

Figura 20: Formulari d'edició de *cables/tubs/fibres*

Aquí podrem afegir *tubs* i *fibres* de forma manual (utilitzant els botons d'*Afegir* o seleccionant una plantilla de pre-establerta)

Template:

Cable 12FO1T TIA598

Editar tubs: 8

Afegir

Editar tub: 1

Eliminar

Color:

black

Fibres:

Afegir

| | | |
|----|--------|----------|
| 3 | grey | Eliminar |
| 2 | orange | Eliminar |
| 4 | green | Eliminar |
| 5 | red | Eliminar |
| 9 | yellow | Eliminar |
| 12 | pink | Eliminar |

| | | |
|----|--------|----------|
| 1 | blue | Eliminar |
| 6 | brown | Eliminar |
| 7 | white | Eliminar |
| 10 | purple | Eliminar |
| 8 | black | Eliminar |
| 11 | cyan | Eliminar |

Actualitzar

Eliminar

Figura 21: Formulari d'edició de *cables/tubs/fibres* amb plantilla

Per tal de guardar els canvis fets haurem de prémer el botó *Update*. Si volem esborrar *fibres/tubs/cable* haurem de prémer els botons d'*Eliminar*.

6.3 Gestió de fusions

Les fusions són les unions entre diferents fibres d'un cable fibra òptica. Per tal de poder realitzar fusions necessitem com a mínim un cable que contingui fibres tot-hi que el cas d'ús habitual és el d'unir diferents *fibres*.

6.3.1 Creació i edició

Per tal de crear *fusions* entre fibres des de la capa de *Xarxa* hem de prémer el botó d'*Editar Fusions*. Aleshores ens apareixerà un formulari amb un llistat de totes les fibres i les relacions entre elles. Per tal de crear *fusions* simplement hem de seleccionar una *fibra* qualsevol i unir-la amb una altra disponible al desplegable.

fiberfy

Caixes

Editar fusions

Gràfic de fusions

Torna entera

Editar fusions: 792

Cable: 8 - cable-625b34d0-d1f4-11e8-b2c6-4b3da9849a1a

Tubs:

black

1

2

3

4

5

6

7

8

9

10

Box- 8.input1

Box- 8.input2

Box- 8.input3

Box- 8.input4

Box- 8.input5

Box- 8.input6

Box- 8.input7

Box- 8.input8

Box- 8.input9

Box- 8.input10

Box- 8.input11

Box- 8.input12

Box- 8.output1

Box- 8.output2

Box- 8.output3

Box- 8.output4

Box- 8.output5

Box- 8.output6

Box- 8.output7

Box- 8.output8

Figura 22: Formulari d'edició de *fusions*

Cable: 8 - cable-625b34d0-d1f4-11e8-b2c6-4b3da9849a1a

Tubs:

black

1

2

3

4

5

6

7

8

9

10

11

Box- 8.input3

Eliminar

Figura 23: Formulari d'edició de *fusions* amb una fusió

Aleshores ens apareixerà un formulari que ens indicarà pels diferents llocs quines *fibres* passen. Aquí podrem seleccionar una a una les fibres que volem unir ⁴.

6.3.2 Visualització

Per visualitzar les fusions realitzades d'un determinat *lloc* s'ha de clicar el botó *Editar fusions* del menú d'infraestructura de xarxa i després seleccionar la pestanya *Gràfic Fusió* del menú superior. Un cop dins es podran veure les unions dels diferents cables amb el seu color corresponent.

⁴A la llista de fibres que apareixerà a cada desplegable, apareixen totes les fibres disponibles i caixes amb (entrades/sortides)

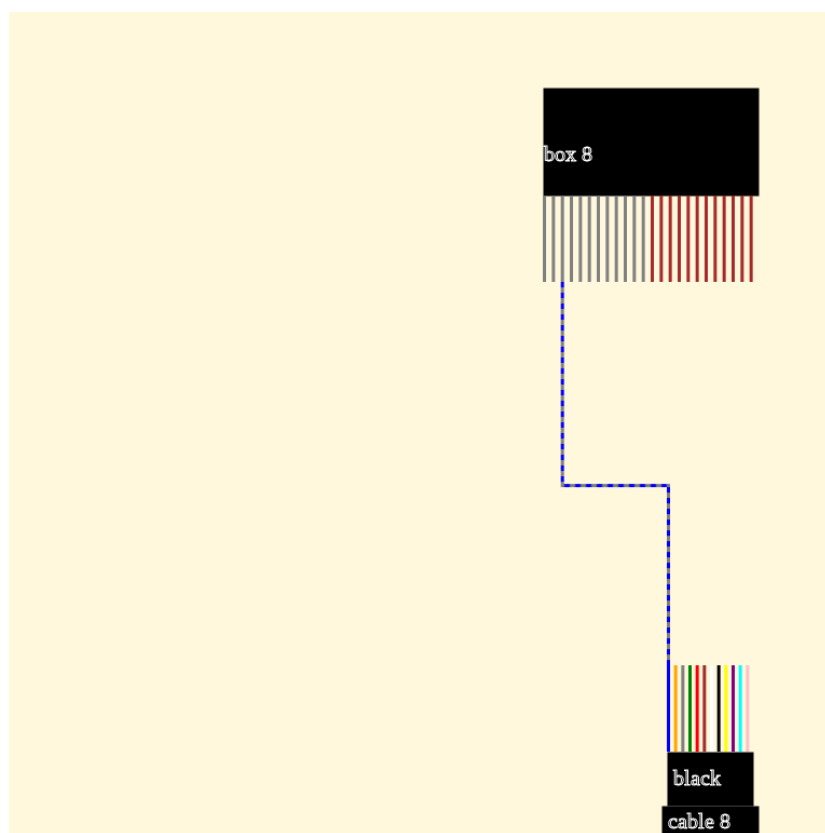


Figura 24: Visor de *fusions*

B Gantt

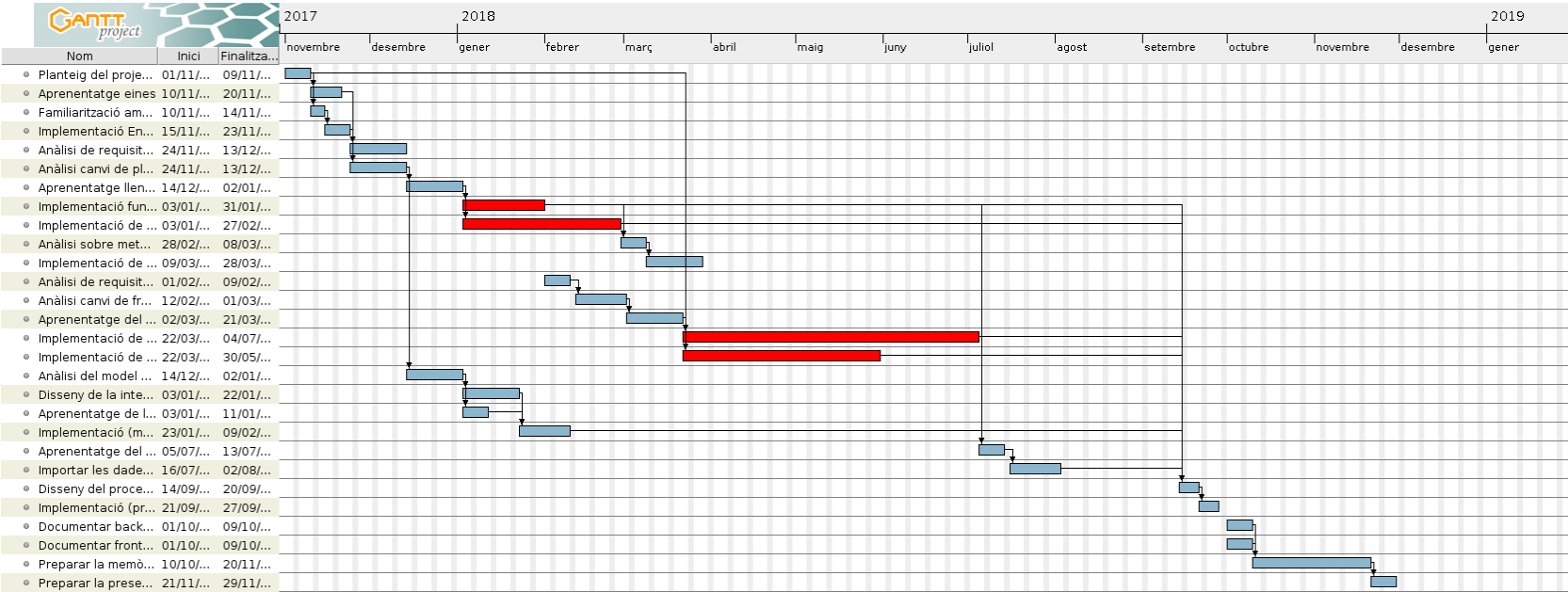


Figura 12.1: Gantt del projecte

C Model de dades

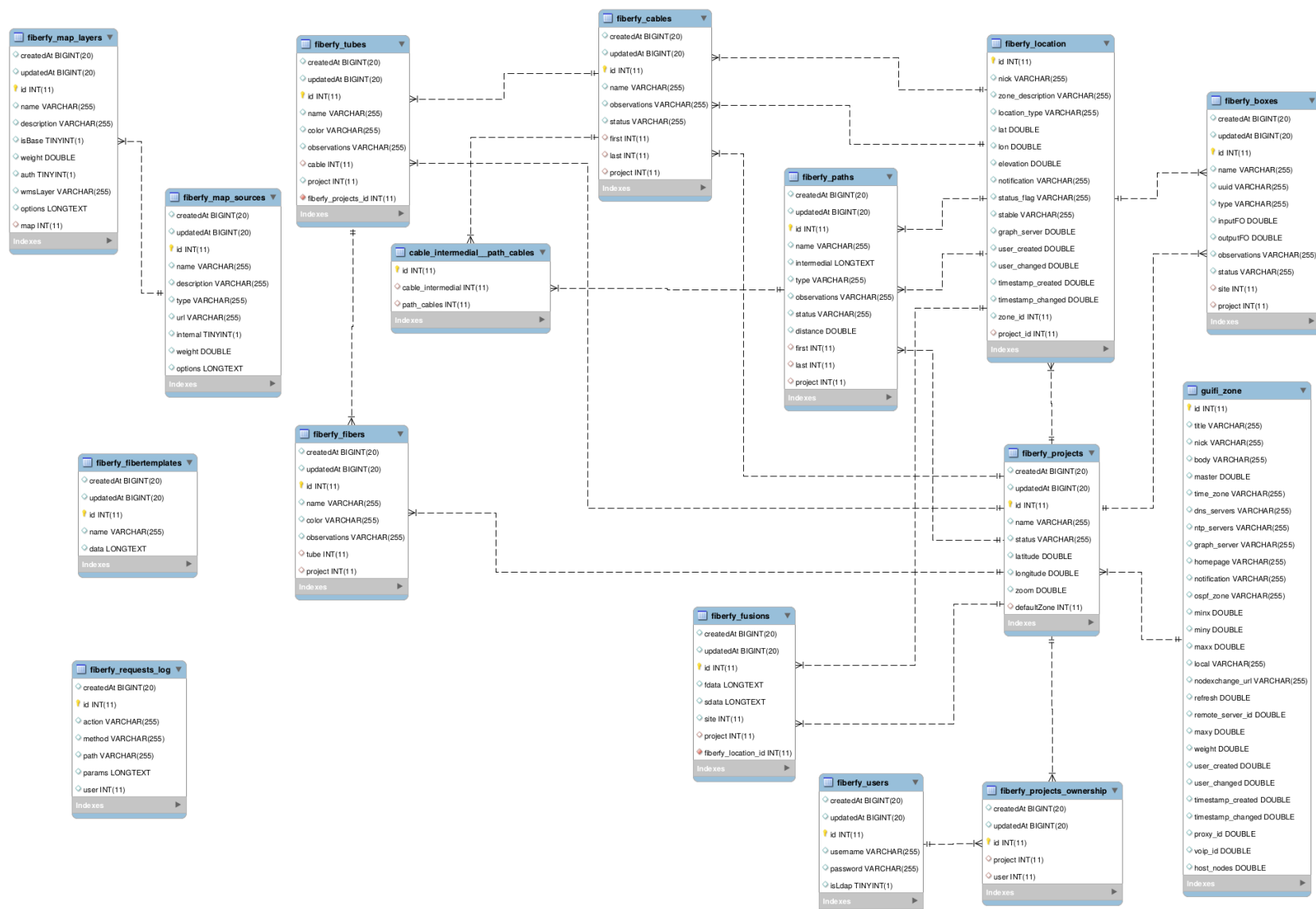


Figura 12.2: Model de dades fiberfy

D Docker images

D.1 docker-fiberfy

Es mostren els fragments més importants de la imatge Docker. Es pot trobar el repositori de forma pública a: <https://github.com/guifi/docker-fiberfy>

Dockerfile

```

1 # Set the base image
2 FROM debian:stretch
3
4 LABEL net.guifi.vendor="Fundacio_guifi.net"
5 LABEL version="1.0"
6 LABEL description="This_docker_image_is_ready_for_\
7 developing_with_guifi.net_fiberfy."
8 LABEL maintainer="roger.garcia@guifi.net"
9
10 ENV NODE_ROOT_DIR /usr/share/node
11 ENV FIBERFY_UNIX_USER fiberfy
12 ENV FIBERFY_USER_ID 1000
13 ENV FIBERFY_USER_GID 1000
14 ENV FIBERFY_GIT_REPO https://github.com/guifi/fiberfy-server.git
15 ENV FIBERFY_GIT_BRANCH master
16
17 RUN apt-get update && apt-get dist-upgrade -y \
18     && apt-get install -y curl wget git gnupg gosu pkg-config \
19     mariadb-client \
20     && apt-get clean \
21     && apt-get autoremove \
22     && rm -rf /var/lib/apt/lists/*
23
24 # Installing nodejs
25 RUN curl -sL https://deb.nodesource.com/setup_8.x | xargs -0 bash -c \
26     && apt-get install -y nodejs
27
28 # Installing sails globally
29 RUN npm install sails -g
30
31 # Installing database waterline-auto tool
32 RUN npm install waterline-auto -g
33
34 # Preparing development dir
35 RUN mkdir -p /usr/share/node/
36
37
38 # Creating UNIX User for development (security reasons)
39 RUN groupadd --gid "${FIBERFY_USER_GID}" "${FIBERFY_UNIX_USER}" && \
40     useradd \
41         --uid ${FIBERFY_USER_ID} \

```

```

42     --gid ${FIBERFY_USER_GID} \
43     --create-home \
44     --shell /bin/bash \
45     ${FIBERFY_UNIX_USER}
46
47 # Define Volume for Drupal
48 VOLUME ${NODE_ROOT_DIR}/fiberfy
49
50
51 # Copy entrypoints
52 COPY ./docker-entrypoint.sh /
53 COPY ./fiberfy-entry.pl /
54
55 # Copying script for setting userid and gid (host)
56 COPY user-mapping.sh /
57 RUN chmod u+x user-mapping.sh
58
59 WORKDIR ${NODE_ROOT_DIR}/fiberfy
60
61 EXPOSE 3000
62 EXPOSE 9229
63
64 ENTRYPOINT ["/docker-entrypoint.sh"]
65
66 CMD ["/bin/bash"]

```

fiberfy-entry.pl

```

1  #!/usr/bin/perl
2  # This script configures fiberfy for first time
3  #
4  use warnings;
5  use strict;
6
7  sleep 15; # We should wait for mariadb container being ready
8
9  print "Checking configurations...\n";
10
11 if (! -e "INSTALLED") {
12
13     my $output = `rm -rf ./ * .[^.]* ..?*`;
14     if ($? != 0) {
15         # Error
16         die "Error erasing all volume.\n";
17     }
18
19     $output = `git init && git remote add origin $ENV{FIBERFY_GIT_REPO} \\\
20               && git fetch && git checkout -t origin/$ENV{
21               FIBERFY_GIT_BRANCH}`;
22     if ($? != 0) {

```

```

22     # Error
23     die "Error_getting_code_from_git_repository.\n";
24 }
25
26 $output = 'chown -R $ENV{FIBERFY_UNIX_USER}.$ENV{FIBERFY_UNIX_USER} .
    && chmod -R o+rw . && chmod -R g+rw .';
27 if ($? != 0) {
28     # Error
29     die "Error_changing_permissions.\n";
30 }
31
32 $output = 'gosu fiberfy ./docker-install.sh';
33 if ($? != 0) {
34     # Error
35     die "Error_running_installation_script.\n";
36 }
37
38
39 # make INSTALLED file
40 $output = 'touch INSTALLED';
41 if ($? != 0) {
42     # Error
43     die "Error_creating_INSTALLED_file.\n";
44 }
45
46 print "Guifi.net_fiberfy_successfully_installed_in_Docker_image!\n";
47 }
48 else {
49     print "Already_installed.\n";
50 }

```

docker-entrpoint.sh

```

1 #!/bin/bash
2 /user-mapping.sh
3 perl /fiberfy-entry.pl
4 /bin/bash -c "$*"

```

user-mapping.sh

```

1 #!/bin/bash
2
3 if [ -z "${FIBERFY_UNIX_USER}" ]; then
4     echo "We_need_USER_to_be_set!"; exit 100
5 fi
6
7 # if both not set we do not need to do anything
8 if [ -z "${HOST_USER_ID}" -a -z "${HOST_USER_GID}" ]; then

```

```

9     echo "Nothing_to_do_here." ; exit 0
10 fi
11
12 # reset user_id to either new id or if empty old (still one of above
13 # might not be set)
14 USER_ID=${HOST_USER_ID:=${FIBERFY_USER_ID}}
15 USER_GID=${HOST_USER_GID:=${FIBERFY_USER_GID}}
16
17 LINE=$(grep -F "${FIBERFY_UNIX_USER}" /etc/passwd)
18 # replace all ':' with a space and create array
19 array=( ${LINE//:/ } )
20
21 # home is 5th element
22 USER_HOME=${array[4]}
23
24 sed -i -e "s/^${FIBERFY_UNIX_USER}:[^:]*\):[0-9]*:[0-9]*/${
    FIBERFY_UNIX_USER}:\ 1:${FIBERFY_USER_ID}:${FIBERFY_USER_GID}/" /etc/
    passwd
25 sed -i -e "s/^${FIBERFY_UNIX_USER}:[^:]*\):[0-9]*/${FIBERFY_UNIX_USER
    }:\ 1:${FIBERFY_USER_GID}/" /etc/group
26
27 chown -R ${FIBERFY_USER_ID}:${FIBERFY_USER_GID} ${USER_HOME}
28
29 exec su - "${FIBERFY_UNIX_USER}"

```

E Docker compositions

E.1 drupal6+guifimaps+fiberfy-sails

```

1 version: "3"
2 services:
3   guifi-drupal:
4     image: guifi/drupal-guifi:drupal6
5     environment:
6       GUIFI_DB: guifidev
7       DRUPAL_ADMIN: admin
8       DRUPAL_ADMIN_PWD: drupal
9       GUIFI_USER_DB: guifi
10      GUIFI_USER_DB_PWD: guifi
11      XDEBUG_PORT: 9000
12      GUIFI_URL: http://dev.guifi.net/static/
13     links:
14       - database
15       - gmaps
16     volumes:
17       - ./guifi-web:/usr/share/drupal/guifi-web/
18     ports:
19       - 8080:80
20     depends_on:

```



```
21     - database
22
23 database:
24     image: "mariadb"
25     environment:
26         MYSQL_ROOT_PASSWORD: admin
27         MYSQL_DATABASE: guifidev
28         MYSQL_USER: guifi
29         MYSQL_PASSWORD: guifi
30     volumes:
31         - ./database:/var/lib/mysql
32
33 myadmin:
34     image: "phpmyadmin/phpmyadmin"
35     links:
36         - database:db
37     ports:
38         - 8000:80
39
40 fiberfy:
41     image: "guifi/fiberfy:sails"
42     environment:
43         FIBERFY_UNIX_USER: fiberfy
44         FIBERFY_GIT_REPO: https://github.com/guifi/fiberfy-server.git
45     command: gosu fiberfy node --inspect=0.0.0.0:9229 app.js
46     links:
47         - database
48     ports:
49         - 1337:1337
50         - 9229:9229
51     volumes:
52         - ./fiberfy-server:/usr/share/node/fiberfy
53
54 gmaps:
55     image: "guifi/guifimaps"
56     environment:
57         GUIFI_WEB: "guifi.net"
58         FIBERFY_API: "fiberfy:1337"
59         FIBERFY_LOGIN: "/auth/loginLDAP"
60         FIBERFY_EXPORT: "/api/v1/export/all"
61         FIBERFY_USERNAME: "${FIBERFY_USERNAME}"
62         FIBERFY_PASSWORD: "${FIBERFY_PASSWORD}"
63     links:
64         - fiberfy
65     ports:
66         - 9090:80
67     volumes:
68         - ./guifimaps:/home/guifi/
69
70 volumes:
71     guifi-web:
72         external: false
```

F fiberfy-server

F.1 Models

Zone

```
1 /**
2  * @module Zone
3  * @description Zone in guifi.net website
4  */
5 module.exports = {
6   migrate: 'safe',
7   tableName: 'guifi_zone',
8   attributes: {
9     updatedAt: false,
10    createdAt: false,
11    title: {
12      type: 'string'
13    },
14    nick: {
15      type: 'string',
16      allowNull: true
17    },
18    body: {
19      type: 'string'
20    },
21    master: {
22      type: 'number'
23    },
24    timeZone: {
25      columnName: 'time_zone',
26      type: 'string'
27    },
28    dnsServers: {
29      columnName: 'dns_servers',
30      type: 'string',
31      allowNull: true
32    },
33    ntpServers: {
34      columnName: 'ntp_servers',
35      type: 'string',
36      allowNull: true
37    },
38    graphServer: {
39      columnName: 'graph_server',
40      type: 'string',
41      allowNull: true
42    },
43    homepage: {
44      type: 'string',
45      allowNull: true
```

```
46 },
47 notification: {
48   type: 'string',
49   defaultsTo: 'guifi@guifi.net'
50 },
51 ospfZone: {
52   columnName: 'ospf_zone',
53   type: 'string',
54   allowNull: true
55 },
56 minx: {
57   type: 'number',
58   allowNull: true
59 },
60 miny: {
61   type: 'number',
62   allowNull: true
63 },
64 maxx: {
65   type: 'number',
66   allowNull: true
67 },
68 local: {
69   type: 'string',
70   defaultsTo: 'Yes'
71 },
72 nodexchangeUrl: {
73   columnName: 'nodexchange_url',
74   type: 'string',
75   allowNull: true
76 },
77 refresh: {
78   type: 'number',
79   allowNull: true
80 },
81 remoteServerId: {
82   columnName: 'remote_server_id',
83   type: 'number',
84   allowNull: true
85 },
86 maxy: {
87   type: 'number',
88   allowNull: true
89 },
90 weight: {
91   type: 'number',
92   defaultsTo: 0
93 },
94 userCreated: {
95   columnName: 'user_created',
96   type: 'number',
97   defaultsTo: 0
```

```
98 },
99   userChanged: {
100     columnName: 'user_changed',
101     type: 'number',
102     defaultsTo: 0
103   },
104   autoCreatedAt: {
105     columnName: 'timestamp_created',
106     type: 'number',
107     defaultsTo: 0
108   },
109   autoUpdatedAt: {
110     columnName: 'timestamp_changed',
111     type: 'number',
112     defaultsTo: 0
113   },
114   proxyId: {
115     columnName: 'proxy_id',
116     type: 'number',
117     defaultsTo: 0
118   },
119   voipId: {
120     columnName: 'voip_id',
121     type: 'number',
122     defaultsTo: 0
123   },
124   hostNodes: {
125     columnName: 'host_nodes',
126     type: 'number',
127     defaultsTo: 0,
128     allowNull: true
129   }
130 },
131 _attributes: {
132   id: {
133     rawType: 'MEDIUMINT(8)_UNSIGNED',
134     type: 'number',
135     primaryKey: true,
136     autoIncrement: true
137   },
138   title: {
139     rawType: 'VARCHAR(255)',
140     type: 'string'
141   },
142   nick: {
143     rawType: 'VARCHAR(10)',
144     type: 'string'
145   },
146   body: {
147     rawType: 'string',
148     type: 'string'
149   },

```

```
150 master: {
151     rawType: 'MEDIUMINT(8)_UNSIGNED',
152     type: 'number'
153 },
154 timeZone: {
155     columnName: 'time_zone',
156     rawType: 'VARCHAR(15)',
157     type: 'string'
158 },
159 dnsServers: {
160     columnName: 'dns_servers',
161     rawType: 'VARCHAR(255)',
162     type: 'string'
163 },
164 ntpServers: {
165     columnName: 'ntp_servers',
166     rawType: 'VARCHAR(255)',
167     type: 'string'
168 },
169 graphServer: {
170     columnName: 'graph_server',
171     rawType: 'VARCHAR(40)',
172     type: 'string',
173     comment: 'Foreign_key_to_guifi_services_(type_SNPGraph)'
174 },
175 homepage: {
176     rawType: 'VARCHAR(255)',
177     type: 'string'
178 },
179 notification: {
180     rawType: 'VARCHAR(1024)',
181     type: 'string',
182     defaultsTo: 'guifi@guifi.net'
183 },
184 ospfZone: {
185     columnName: 'ospf_zone',
186     rawType: 'VARCHAR(255)',
187     type: 'string'
188 },
189 minx: {
190     rawType: 'DECIMAL(10,6)',
191     type: 'number'
192 },
193 miny: {
194     rawType: 'DECIMAL(10,6)',
195     type: 'number'
196 },
197 maxx: {
198     rawType: 'DECIMAL(10,6)',
199     type: 'number'
200 },
201 local: {
```

```
202     rawType: 'VARCHAR(5)',
203     type: 'string',
204     defaultsTo: 'Yes',
205     comment: 'Yes,No'
206   },
207   nodexchangeUrl: {
208     columnName: 'nodexchange_url',
209     rawType: 'VARCHAR(255)',
210     type: 'string'
211   },
212   refresh: {
213     rawType: 'INT(11)',
214     type: 'number'
215   },
216   remoteServerId: {
217     columnName: 'remote_server_id',
218     rawType: 'MEDIUMINT(9)',
219     type: 'number'
220   },
221   maxy: {
222     rawType: 'DECIMAL(10,6)',
223     type: 'number'
224   },
225   weight: {
226     rawType: 'TINYINT(4)',
227     type: 'number',
228     defaultsTo: 0
229   },
230   userCreated: {
231     columnName: 'user_created',
232     rawType: 'MEDIUMINT(9)',
233     type: 'number',
234     defaultsTo: 0
235   },
236   userChanged: {
237     columnName: 'user_changed',
238     rawType: 'MEDIUMINT(9)',
239     type: 'number',
240     defaultsTo: 0
241   },
242   autoCreatedAt: {
243     columnName: 'timestamp_created',
244     rawType: 'INT(11)',
245     type: 'number',
246     defaultsTo: 0
247   },
248   autoUpdatedAt: {
249     columnName: 'timestamp_changed',
250     rawType: 'INT(11)',
251     type: 'number',
252     defaultsTo: 0
253   },
```

```
254 proxyId: {
255   columnName: 'proxy_id',
256   rawType: 'MEDIUMINT(9)',
257   type: 'number',
258   defaultsTo: 0
259 },
260 voipId: {
261   columnName: 'voip_id',
262   rawType: 'MEDIUMINT(9)',
263   type: 'number',
264   defaultsTo: 0
265 },
266 hostNodes: {
267   columnName: 'host_nodes',
268   rawType: 'SMALLINT(6)',
269   type: 'number',
270   defaultsTo: 0
271 }
272 }
273 }
```

Site

```
1 /**
2  * @module Site
3  * @description Site in a civil work
4  */
5
6 module.exports = {
7   migrate: 'safe',
8   tableName: 'fiberfy_location',
9   _swagger: {
10     attributes: {
11       ignore: {
12         request: {
13           id: true,
14           autoUpdatedAt: true,
15           autoCreatedAt: true
16         },
17         response: {
18
19         }
20       },
21       append: {
22         request: {
23           type: {
24             enum: [
25               'notdefined',
26               'manhole',
27               'pole',
```

```
28         'room',
29         'cabinet',
30         'poe',
31         'hook',
32         'jump'
33     ]
34 }
35 },
36 response: {
37     type: {
38         enum: [
39             'notdefined',
40             'manhole',
41             'pole',
42             'room',
43             'cabinet',
44             'poe',
45             'hook',
46             'jump'
47         ]
48     }
49 }
50 }
51 }
52 },
53 attributes: {
54     updatedAt: false,
55     createdAt: false,
56     name: {
57         columnName: 'nick',
58         type: 'string'
59     },
60     zone: {
61         columnName: 'zone_id',
62         model: 'Zone'
63     },
64     observations: {
65         columnName: 'zone_description',
66         type: 'string',
67         allowNull: true
68     },
69     type: {
70         columnName: 'location_type',
71         type: 'string',
72         defaultsTo: 'notdefined',
73         /* isIn: [
74             'notdefined',
75             'manhole',
76             'pole',
77             'room',
78             'cabinet',
79             'poe',
```



```
80         'hook',
81         'jump'
82     ] */
83 },
84 latitude: {
85     columnName: 'lat',
86     type: 'number',
87     allowNull: true
88 },
89 longitude: {
90     columnName: 'lon',
91     type: 'number',
92     allowNull: true
93 },
94 elevation: {
95     type: 'number',
96     allowNull: true
97 },
98 notification: {
99     type: 'string',
100     defaultsTo: 'guifi@guifi.net'
101 },
102 status: {
103     columnName: 'status_flag',
104     type: 'string',
105     defaultsTo: 'Planned'
106 },
107 project: {
108     columnName: 'project_id',
109     model: 'Project'
110 },
111 stable: {
112     type: 'string',
113     defaultsTo: 'Yes'
114 },
115 graphServer: {
116     columnName: 'graph_server',
117     type: 'number',
118     defaultsTo: 0
119 },
120 userCreated: {
121     columnName: 'user_created',
122     type: 'number',
123     defaultsTo: 1
124 },
125 userChanged: {
126     columnName: 'user_changed',
127     type: 'number',
128     allowNull: true
129 },
130 autoCreatedAt: {
131     columnName: 'timestamp_created',
```

```
132     type: 'number',
133     defaultsTo: 0
134   },
135   autoUpdatedAt: {
136     columnName: 'timestamp_changed',
137     type: 'number',
138     allowNull: true
139   }
140 },
141 _attributes: {
142   id: {
143     rawType: 'MEDIUMINT(9)',
144     type: 'number',
145     primaryKey: true,
146     autoIncrement: true
147   },
148   name: {
149     rawType: 'VARCHAR(40)',
150     type: 'string'
151   },
152   zone: {
153     columnName: 'zone_id',
154     rawType: 'MEDIUMINT(9)',
155     type: 'number',
156     defaultsTo: 2413
157   },
158   observations: {
159     columnName: 'zone_description',
160     rawType: 'VARCHAR(255)',
161     type: 'string'
162   },
163   type: {
164     columnName: 'location_type',
165     rawType: 'VARCHAR(10)',
166     type: 'string',
167     defaultsTo: 'node'
168   },
169   latitude: {
170     rawType: 'DECIMAL(10,6)',
171     type: 'number'
172   },
173   longitude: {
174     rawType: 'DECIMAL(10,6)',
175     type: 'number'
176   },
177   elevation: {
178     rawType: 'TINYINT(4)',
179     type: 'number'
180   },
181   notification: {
182     rawType: 'VARCHAR(1024)',
183     type: 'string',
```

```
184     defaultsTo: 'guifi@guifi.net'
185   },
186   status: {
187     columnName: 'status_flag',
188     rawType: 'VARCHAR(40)',
189     type: 'string',
190     defaultsTo: 'Planned'
191   },
192   project: {
193     columnName: 'project_id',
194     rawType: 'INT(11)',
195     type: 'number'
196   },
197   stable: {
198     rawType: 'VARCHAR(25)',
199     type: 'string',
200     defaultsTo: 'Yes'
201   },
202   graphServer: {
203     columnName: 'graph_server',
204     rawType: 'MEDIUMINT(9)',
205     type: 'number',
206     defaultsTo: 0,
207     comment: 'Foreign_key_to_guifi_services_(type_SNPGraph)'
208   },
209   userCreated: {
210     columnName: 'user_created',
211     rawType: 'MEDIUMINT(9)',
212     type: 'number',
213     defaultsTo: 0
214   },
215   userChanged: {
216     columnName: 'user_changed',
217     rawType: 'MEDIUMINT(9)',
218     type: 'number'
219   },
220   autoCreatedAt: {
221     columnName: 'timestamp_created',
222     rawType: 'INT(11)',
223     type: 'number',
224     defaultsTo: 0
225   },
226   autoUpdatedAt: {
227     columnName: 'timestamp_changed',
228     rawType: 'INT(11)',
229     type: 'number'
230   },
231   toJSON: function () {
232     const model = this.toObject();
233     if (model.type === 'node') return false;
234     return model;
235   }
```

```
236 },
237 beforeCreate : function (values , next) {
238     var seconds = Math.floor(new Date() / 1000);
239     values.autoCreatedAt = seconds;
240     values.userCreated = 1; // Hardcodegem id d'usuari (testing)
241     values.graphServer = 0;
242     values.notification = 'guifi@guifi.net';
243     next();
244 },
245 beforeUpdate : function (values , next) {
246     var seconds = Math.floor(new Date() / 1000);
247     values.autoUpdatedAt = seconds;
248     next();
249 },
250 beforeDestroy : function (values , next) {
251     if (values.where.id) {
252         Path.destroy({or: [{first: values.where.id}, {last: values.where.id
253             }]}).exec(function (err) {
254             next(err);
255         })
256     } else {
257         next();
258     }
259 };
```

Project

```
1 /**
2  * @module Project
3  * @description Workspace group by users
4  */
5
6 module.exports = {
7     tableName: 'fiberfy_projects',
8     cascadeOnDestroy: true,
9     _swagger: {
10         attributes: {
11             ignore: {
12                 request: {
13                     id: true,
14                     updatedAt: true,
15                     createdAt: true,
16                 },
17                 response: {
18
19                 }
20             },
21             append: {
22                 request: {
```

```
23     },
24     response: {
25     },
26   },
27   additions: {
28     request: {
29       defaultZone: {
30         type: 'integer'
31       }
32     },
33     response: {
34       defaultZone: {
35         type: 'integer'
36       }
37     }
38   }
39 },
40 },
41 attributes: {
42   name: {
43     type: 'string'
44   },
45   status: {
46     type: 'string'
47   },
48   latitude: {
49     type: 'number'
50   },
51   longitude: {
52     type: 'number'
53   },
54   zoom: {
55     type: 'number'
56   },
57   defaultZone : {
58     model: 'Zone'
59   },
60   users: {
61     collection: 'User',
62     via: 'project',
63     through: 'ProjectOwnership'
64   }
65 },
66 beforeDestroy : async function (criteria , proceed) {
67   if (criteria.where.id) {
68     try {
69       await Site.destroy({project: criteria.where.id})
70       await Path.destroy({project: criteria.where.id})
71       await Cable.destroy({project: criteria.where.id})
72       await Box.destroy({project: criteria.where.id})
73       proceed();
74     } catch (err) {
```

```

75     proceed(err);
76   }
77   } else {
78     proceed();
79   }
80 }
81 };

```

Path

```

1  /**
2   * @module Path
3   * @description Path between Site in a civil work
4   */
5
6  module.exports = {
7    tableName: 'fiberfy_paths',
8    _swagger: {
9      attributes: {
10        ignore: {
11          request: {
12            id: true,
13            updatedAt: true,
14            createdAt: true,
15            distance: true,
16            intermedial: true
17          },
18          response: {
19            intermedial: true
20          }
21        },
22        append: {
23          request: {
24            type: {
25              enum: [
26                'notdefined',
27                'underground',
28                'facade',
29                'aerial'
30              ]
31            }
32          },
33          response: {
34            type: {
35              enum: [
36                'notdefined',
37                'underground',
38                'facade',
39                'aerial'
40              ]

```

```
41     }
42   }
43 },
44 additions: {
45   request: {
46     first: {
47       type: 'integer'
48     },
49     last: {
50       type: 'integer'
51     },
52     intermedial: {
53       type: 'array',
54       items: {
55         type: 'array',
56         items: {
57           type: 'integer'
58         },
59         minItems: 2,
60         maxItems: 2
61       }
62     }
63   },
64   response: {
65     intermedial: {
66       type: 'array',
67       items: {
68         type: 'array',
69         items: {
70           type: 'integer'
71         },
72         minItems: 2,
73         maxItems: 2
74       }
75     }
76   }
77 }
78 },
79 attributes: {
80   name: {
81     type: 'string'
82   },
83   first: {
84     model: 'Site'
85   },
86   last: {
87     model: 'Site'
88   },
89   intermedial: {
90     type: 'json'
91   },
92 }
```

```
93   type: {
94     type: 'string'
95   },
96   observations: {
97     type: 'string'
98   },
99   status: {
100    type: 'string'
101  },
102  distance: {
103    type: 'number',
104    required: true
105  },
106  cables: {
107    collection: 'Cable',
108    via: 'intermedial'
109  },
110  project: {
111    model: 'Project'
112  }
113 }
114 };
```

Box

```
1 /**
2  * @module Site
3  * @description Site in a civil work
4  */
5
6 module.exports = {
7   migrate: 'safe',
8   tableName: 'fiberfy_location',
9   _swagger: {
10     attributes: {
11       ignore: {
12         request: {
13           id: true,
14           autoUpdatedAt: true,
15           autoCreatedAt: true
16         },
17         response: {
18
19         }
20       },
21       append: {
22         request: {
23           type: {
24             enum: [
25               'notdefined',
```



```
26         'manhole',
27         'pole',
28         'room',
29         'cabinet',
30         'poe',
31         'hook',
32         'jump'
33     ]
34 }
35 },
36 response: {
37     type: {
38         enum: [
39             'notdefined',
40             'manhole',
41             'pole',
42             'room',
43             'cabinet',
44             'poe',
45             'hook',
46             'jump'
47         ]
48     }
49 }
50 }
51 },
52 },
53 attributes: {
54     updatedAt: false,
55     createdAt: false,
56     name: {
57         columnName: 'nick',
58         type: 'string'
59     },
60     zone: {
61         columnName: 'zone_id',
62         model: 'Zone'
63     },
64     observations: {
65         columnName: 'zone_description',
66         type: 'string',
67         allowNull: true
68     },
69     type: {
70         columnName: 'location_type',
71         type: 'string',
72         defaultsTo: 'notdefined',
73         /* isIn: [
74             'notdefined',
75             'manhole',
76             'pole',
77             'room',
```

```
78         'cabinet',
79         'poe',
80         'hook',
81         'jump'
82     ] */
83 },
84 latitude: {
85     columnName: 'lat',
86     type: 'number',
87     allowNull: true
88 },
89 longitude: {
90     columnName: 'lon',
91     type: 'number',
92     allowNull: true
93 },
94 elevation: {
95     type: 'number',
96     allowNull: true
97 },
98 notification: {
99     type: 'string',
100     defaultsTo: 'guifi@guifi.net'
101 },
102 status: {
103     columnName: 'status_flag',
104     type: 'string',
105     defaultsTo: 'Planned'
106 },
107 project: {
108     columnName: 'project_id',
109     model: 'Project'
110 },
111 stable: {
112     type: 'string',
113     defaultsTo: 'Yes'
114 },
115 graphServer: {
116     columnName: 'graph_server',
117     type: 'number',
118     defaultsTo: 0
119 },
120 userCreated: {
121     columnName: 'user_created',
122     type: 'number',
123     defaultsTo: 1
124 },
125 userChanged: {
126     columnName: 'user_changed',
127     type: 'number',
128     allowNull: true
129 },
```

```
130     autoCreatedAt: {
131       columnName: 'timestamp_created',
132       type: 'number',
133       defaultsTo: 0
134     },
135     autoUpdatedAt: {
136       columnName: 'timestamp_changed',
137       type: 'number',
138       allowNull: true
139     }
140   },
141   _attributes: {
142     id: {
143       rawType: 'MEDIUMINT(9)',
144       type: 'number',
145       primaryKey: true,
146       autoIncrement: true
147     },
148     name: {
149       rawType: 'VARCHAR(40)',
150       type: 'string'
151     },
152     zone: {
153       columnName: 'zone_id',
154       rawType: 'MEDIUMINT(9)',
155       type: 'number',
156       defaultsTo: 2413
157     },
158     observations: {
159       columnName: 'zone_description',
160       rawType: 'VARCHAR(255)',
161       type: 'string'
162     },
163     type: {
164       columnName: 'location_type',
165       rawType: 'VARCHAR(10)',
166       type: 'string',
167       defaultsTo: 'node'
168     },
169     latitude: {
170       rawType: 'DECIMAL(10,6)',
171       type: 'number'
172     },
173     longitude: {
174       rawType: 'DECIMAL(10,6)',
175       type: 'number'
176     },
177     elevation: {
178       rawType: 'TINYINT(4)',
179       type: 'number'
180     },
181     notification: {
```

```
182     rawType: 'VARCHAR(1024)',
183     type: 'string',
184     defaultsTo: 'guifi@guifi.net'
185   },
186   status: {
187     columnName: 'status_flag',
188     rawType: 'VARCHAR(40)',
189     type: 'string',
190     defaultsTo: 'Planned'
191   },
192   project: {
193     columnName: 'project_id',
194     rawType: 'INT(11)',
195     type: 'number'
196   },
197   stable: {
198     rawType: 'VARCHAR(25)',
199     type: 'string',
200     defaultsTo: 'Yes'
201   },
202   graphServer: {
203     columnName: 'graph_server',
204     rawType: 'MEDIUMINT(9)',
205     type: 'number',
206     defaultsTo: 0,
207     comment: 'Foreign_key_to_guifi_services_(type_SNPGraph)'
208   },
209   userCreated: {
210     columnName: 'user_created',
211     rawType: 'MEDIUMINT(9)',
212     type: 'number',
213     defaultsTo: 0
214   },
215   userChanged: {
216     columnName: 'user_changed',
217     rawType: 'MEDIUMINT(9)',
218     type: 'number'
219   },
220   autoCreatedAt: {
221     columnName: 'timestamp_created',
222     rawType: 'INT(11)',
223     type: 'number',
224     defaultsTo: 0
225   },
226   autoUpdatedAt: {
227     columnName: 'timestamp_changed',
228     rawType: 'INT(11)',
229     type: 'number'
230   },
231   toJSON: function () {
232     const model = this.toObject();
233     if (model.type === 'node') return false;
```

```

234     return model;
235   }
236 },
237 beforeCreate : function (values , next) {
238   var seconds = Math.floor(new Date() / 1000);
239   values.autoCreatedAt = seconds;
240   values.userCreated = 1; // Hardcodegem id d'usuari (testing)
241   values.graphServer = 0;
242   values.notification = 'guifi@guifi.net';
243   next();
244 },
245 beforeUpdate : function (values , next) {
246   var seconds = Math.floor(new Date() / 1000);
247   values.autoUpdatedAt = seconds;
248   next();
249 },
250 beforeDestroy : function (values , next) {
251   if (values.where.id) {
252     Path.destroy({or: [{first: values.where.id}, {last: values.where.id
253       }]}).exec(function (err) {
254       next(err);
255     })
256   } else {
257     next();
258   }
259 };

```

Cable, tube, fiber

```

1  /**
2   * @module Cable
3   * @description Cable between Site, and intermedial == site_id intermedial.
4   */
5
6  module.exports = {
7    tableName: 'fiberfy_cables',
8    attributes: {
9      name: {
10        type: 'string'
11      },
12      first: {
13        model: 'Site',
14        required: true
15      },
16      last: {
17        model: 'Site',
18        required: true
19      },
20      intermedial: {

```

```

21     collection: 'Path',
22     via: 'cables',
23     /* type: 'json',
24     required: true */
25   },
26   tubes: {
27     collection: 'Tube',
28     via: 'cable'
29   },
30   observations: {
31     type: 'string'
32   },
33   status: {
34     type: 'string'
35   },
36
37   project: {
38     model: 'Project'
39   }
40 },
41 beforeDestroy : function (criteria , proceed) {
42   if (criteria.where.id) {
43     Tube.destroy({cable: criteria.where.id}).exec(function(err) {
44       proceed(err);
45     })
46   } else {
47     proceed();
48   }
49 }
50 };

```

```

1 /**
2  * @module Tube
3  * @description Tube inside Cable
4  */
5
6 module.exports = {
7   tableName: 'fiberfy_tubes',
8   attributes: {
9     name: {
10       type: 'string'
11     },
12     cable: {
13       model: 'Cable',
14       required: true
15     },
16     color: {
17       type: 'string'
18     },
19     fibers: {
20       collection: 'Fiber',
21       via: 'tube'

```

```
22     },
23     observations: {
24       type: 'string'
25     },
26
27     project: {
28       model: 'Project'
29     }
30   },
31   beforeDestroy : function (criteria , proceed) {
32     if (criteria.where.id) {
33       Fiber.destroy({tube: criteria.where.id}).exec(function(err) {
34         proceed(err);
35       })
36     } else {
37       proceed();
38     }
39   }
40 };
```

```
1 /**
2  * @module Fiber
3  * @description Fiber inside Tube
4  */
5
6 module.exports = {
7   tableName: 'fiberfy_fibers',
8   attributes: {
9     name: {
10       type: 'string'
11     },
12     tube: {
13       model: 'Tube',
14       required: true
15     },
16     color: {
17       type: 'string'
18     },
19     observations: {
20       type: 'string'
21     },
22
23     project: {
24       model: 'Project'
25     }
26   }
27 };
```

F.2 Serveis

ldap

```

1  /**
2   * ldap
3   *
4   * @description :: LDAP service for authenticating users
5   * @help        :: See https://github.com/auth0/node-jsonwebtoken & http://
6   *               sailsjs.org/#!/documentation/concepts/Services
7   */
8  const ldapjs = require('ldapjs');
9  const assert = require('assert');
10
11 // Generates a token from supplied payload
12 module.exports.search = function(username, callback) {
13
14   const ldap_user = process.env.LDAP_USER
15   const ldap_pass = process.env.LDAP_PASSWORD
16
17   var data = {};
18
19   process.env.NODE_TLS_REJECT_UNAUTHORIZED = "0";
20   var client = ldapjs.createClient({
21     url: 'ldaps://ldap.guifi.net',
22   });
23
24   client.on('error', function(err) {
25     callback({ error: err });
26     console.log(err);
27   });
28
29   client.bind('uid=' + ldap_user + ',o=glirusers,dc=guifi,dc=net',
30     ldap_pass, function(err) { if (err) {
31       if (err.lde_message == "Invalid_Credentials" ){
32         callback ({ error: err.lde_message })
33       }
34     } else {
35       var opts = {
36         filter: '(&(cn=lectura)(memberUid=' + username + '))',
37         scope: 'sub',
38         attributes: ['cn']
39       };
40
41       client.search('o=glirusers,dc=guifi,dc=net', opts, function(err, res)
42         {
43           assert.ifError(err);
44
45           res.on('searchEntry', function(entry) {
46             data = entry.object;
47           });
48         }
49       );
50     }
51   });
52 }

```



```

47     res.on('error', function(err) {
48         console.error('error:_' + err.message);
49         callback ({ error: err.lde_message })
50     });
51     res.on('end', function(result) {
52         callback(data);
53     });
54
55 });
56 }
57 });
58
59 }
60
61 // Verifies token on a request
62 module.exports.authenticate = function(credentials, callback) {
63     const ldap_user = process.env.LDAP_USER
64     const ldap_pass = process.env.LDAP_PASSWORD
65
66
67     process.env.NODE_TLS_REJECT_UNAUTHORIZED = "0";
68     var client = ldapjs.createClient({
69         url: 'ldaps://ldap.guifi.net',
70     });
71
72     client.on('error', function(err) {
73         callback({ error: err });
74         console.log(err);
75     });
76
77     client.bind('uid=' + credentials.username + ',o=glirusers,dc=guifi,dc=net',
78         credentials.password, function(err) { if (err) {
79         if (err.lde_message === "Invalid_Credentials" ) {
80             console.log("Wrong_Credentials!");
81             callback({ error: err.lde_message });
82         } else {
83             console.log(err);
84             callback({ error: err.lde_message });
85         }
86     } else {
87         callback({ message: 'Good_Credentials' })
88     }
89 });
90 };

```

F.3 API

Especificació

```
1 openapi: 3.0.0
2 info:
3   version: '1.0'
4   title: fiberfy-server backend
5 paths:
6   /auth/loginLDAP:
7     post:
8       description: Login using LDAP and Returns a JWT token to
9         the caller
10      operationId: authLoginLDAP
11      tags:
12        - Authentication
13      requestBody:
14        content:
15          application/json:
16            schema:
17              type: object
18              properties:
19                username:
20                  type: string
21                  description: LDAP username
22                password:
23                  type: string
24                  description: LDAP password
25      responses:
26        '200':
27          description: Success
28          content:
29            application/json:
30              schema:
31                $ref: '#/components/schemas/LoginResponse'
32        default:
33          description: Error
34          content:
35            application/json:
36              schema:
37                $ref: '#/components/schemas/ErrorResponse'
38   /auth/login:
39     post:
40       description: Login using local authentication and Returns
41         a JWT token to the caller
42       operationId: authLoginLocal
43       tags:
```

```

42     - Authentication
43     requestBody:
44       content:
45         application/json:
46           schema:
47             type: object
48             properties:
49               username:
50                 type: string
51                 description: LDAP username
52               password:
53                 type: string
54                 description: LDAP password
55     responses:
56       '200':
57         description: Success
58         content:
59           application/json:
60             schema:
61               $ref: '#/components/schemas/LoginResponse'
62       default:
63         description: Error
64         content:
65           application/json:
66             schema:
67               $ref: '#/components/schemas/ErrorResponse'
68 /import:
69   post:
70     description: ➤
71     Import geographical data (PolyLines and Points) as
       civil infrastructure
72     (Sites , Paths)
73     operationId: import
74     tags:
75       - I/O
76     security:
77       - BearerAuth: []
78     requestBody:
79       content:
80         multipart/form-data:
81           schema:
82             type: object
83             properties:

```

```

84         data:
85             type: string
86             format: binary
87             description: 'The_file_to_upload._In_format_
                        KML,_KMZ_or_GeoJSON'
88         project:
89             type: integer
90             description: Project id to import data
91         defaultZone:
92             type: integer
93             description: Zone to import data
94         threshold:
95             type: integer
96             description: >
97                 Threshold to attach near paths to the same
98                 site in meters.
99                 By default is 10 meters.
100     responses:
101         '200':
102             description: Success
103             content:
104                 application/json:
105                     schema:
106                         $ref: '#/components/schemas/ImportResponse'
107             default:
108                 description: Error
109                 content:
110                     application/json:
111                         schema:
112                             $ref: '#/components/schemas/ErrorResponse'
113 /export/{id}:
114     get:
115         description: >
116             Export project civil infrastructure (geographical data)
117             as PolyLines and Points
118             (Sites, Paths) in GeoJSON format
119         operationId: export
120         tags:
121             - I/O
122         security:
123             - BearerAuth: []
124         parameters:
125             - in: path

```

```

124         name: id
125         description: Project id
126         schema:
127             type: integer
128     responses:
129         '200':
130             description: Success
131             content:
132                 application/json:
133                     schema:
134                         $ref: '#/components/schemas/ExportResponse'
135         default:
136             description: Error
137             content:
138                 application/json:
139                     schema:
140                         $ref: '#/components/schemas/ErrorResponse'
141 /export/all:
142     get:
143         description: ➤
144             Export all civil infrastructure (geographical data) as
145             PolyLines and Points
146             (Sites , Paths) in GeoJSON format
147         operationId: export
148         tags:
149             - I/O
150         security:
151             - BearerAuth: []
152         responses:
153             '200':
154                 description: Success
155                 content:
156                     application/json:
157                         schema:
158                             $ref: '#/components/schemas/ExportResponse'
159             default:
160                 description: Error
161                 content:
162                     application/json:
163                         schema:
164                             $ref: '#/components/schemas/ErrorResponse'
165 /site:
166     post:

```

```
166     description: Post a Site (Locations which stores one
167           point of civil infrastructure)
168     operationId: sitePost
169     tags:
170       - Site
171     security:
172       - BearerAuth: []
173     requestBody:
174       content:
175         application/json:
176           schema:
177             type: object
178             $ref: 'models/Site.json#/request'
179     responses:
180       '200':
181         description: Success
182         content:
183           application/json:
184             schema:
185               $ref: 'models/Site.json#/response'
186       default:
187         description: Error
188         content:
189           application/json:
190             schema:
191               $ref: '#/components/schemas/ErrorResponse'
192     get:
193       description: Query Site model
194       operationId: siteGet
195       tags:
196         - Site
197       security:
198         - BearerAuth: []
199       responses:
200         '200':
201           description: Success
202           content:
203             application/json:
204               schema:
205                 type: array
206                 items:
207                   type: object
208                   $ref: 'models/Site.json#/response'
```

```
208     default:
209       description: Error
210       content:
211         application/json:
212           schema:
213             $ref: '#/components/schemas/ErrorResponse'
214 /site/{id}:
215   patch:
216     description: Update a Site (Locations which stores one
217       point of civil infrastructure)
218     operationId: sitePatch
219     tags:
220       - Site
221     security:
222       - BearerAuth: []
223     parameters:
224       - in: path
225         name: id
226         schema:
227           type: integer
228     requestBody:
229       content:
230         application/json:
231           schema:
232             type: object
233             $ref: 'models/Site.json#/request'
234     responses:
235       '200':
236         description: Success
237         content:
238           application/json:
239             schema:
240               $ref: 'models/Site.json#/response'
241       default:
242         description: Error
243         content:
244           application/json:
245             schema:
246               $ref: '#/components/schemas/ErrorResponse'
247   delete:
248     description: Destroy a Site (Locations which stores one
249       point of civil infrastructure)
```

```
249     operationId: siteDelete
250     tags:
251       - Site
252     security:
253       - BearerAuth: []
254     parameters:
255       - in: path
256         name: id
257         schema:
258           type: integer
259     responses:
260       '200':
261         description: Success
262         content:
263           application/json:
264             schema:
265               $ref: 'models/Site.json#/response'
266       default:
267         description: Error
268         content:
269           application/json:
270             schema:
271               $ref: '#/components/schemas/ErrorResponse'
272   get:
273     description: Get given Site information (Locations which
274                 stores one point of civil infrastructure)
275     operationId: siteGetOne
276     tags:
277       - Site
278     security:
279       - BearerAuth: []
280     parameters:
281       - in: path
282         name: id
283         schema:
284           type: integer
285     responses:
286       '200':
287         description: Success
288         content:
289           application/json:
290             schema:
291               $ref: 'models/Site.json#/response'
```



```
291     default:
292       description: Error
293       content:
294         application/json:
295           schema:
296             $ref: '#/components/schemas/ErrorResponse'
297 /path:
298   post:
299     description: Post a Path (Civil work paths between Sites)
300     operationId: pathPost
301     tags:
302       - Path
303     security:
304       - BearerAuth: []
305     requestBody:
306       content:
307         application/json:
308           schema:
309             type: object
310             $ref: 'models/Path.json#/request'
311     responses:
312       '200':
313         description: Success
314         content:
315           application/json:
316             schema:
317               $ref: 'models/Path.json#/response'
318         default:
319           description: Error
320           content:
321             application/json:
322               schema:
323                 $ref: '#/components/schemas/ErrorResponse'
324   get:
325     description: Query Path model
326     operationId: pathGet
327     tags:
328       - Path
329     security:
330       - BearerAuth: []
331     responses:
332       '200':
333         description: Success
```

```
334         content:
335           application/json:
336             schema:
337               type: array
338             items:
339               type: object
340               $ref: 'models/Path.json#/response'
341         default:
342           description: Error
343           content:
344             application/json:
345               schema:
346                 $ref: '#/components/schemas/ErrorResponse'
347 /path/{id}:
348   patch:
349     description: Update a Path (Civil work paths between
350                Sites)
351     operationId: pathPatch
352     tags:
353       - Path
354     security:
355       - BearerAuth: []
356     parameters:
357       - in: path
358         name: id
359         schema:
360           type: integer
361     requestBody:
362       content:
363         application/json:
364           schema:
365             type: object
366             $ref: 'models/Path.json#/request'
367     responses:
368       '200':
369         description: Success
370         content:
371           application/json:
372             schema:
373               $ref: 'models/Path.json#/response'
374         default:
375           description: Error
376           content:
```

```

376         application/json:
377             schema:
378                 $ref: '#/components/schemas/ErrorResponse'
379
380     delete:
381         description: Destroy a Path (Civil work paths between
382             Sites)
383         operationId: pathDelete
384         tags:
385             - Path
386         security:
387             - BearerAuth: []
388         parameters:
389             - in: path
390               name: id
391               schema:
392                   type: integer
393         responses:
394             '200':
395                 description: Success
396                 content:
397                     application/json:
398                         schema:
399                             $ref: '#/models/Path.json#/response'
400             default:
401                 description: Error
402                 content:
403                     application/json:
404                         schema:
405                             $ref: '#/components/schemas/ErrorResponse'
406     get:
407         description: Get given Path information (Civil work paths
408             between Sites)
409         operationId: pathGetOne
410         tags:
411             - Path
412         security:
413             - BearerAuth: []
414         parameters:
415             - in: path
416               name: id
417               schema:
418                   type: integer

```

```

417     responses:
418       '200':
419         description: Success
420         content:
421           application/json:
422             schema:
423               $ref: 'models/Path.json#/response'
424       default:
425         description: Error
426         content:
427           application/json:
428             schema:
429               $ref: '#/components/schemas/ErrorResponse'
430
431 /project:
432   post:
433     description: Post a Project
434     operationId: projectPost
435     tags:
436       - Project
437     security:
438       - BearerAuth: []
439     requestBody:
440       content:
441         application/json:
442           schema:
443             type: object
444             $ref: 'models/Project.json#/request'
445     responses:
446       '200':
447         description: Success
448         content:
449           application/json:
450             schema:
451               $ref: 'models/Project.json#/response'
452       default:
453         description: Error
454         content:
455           application/json:
456             schema:
457               $ref: '#/components/schemas/ErrorResponse'
458   get:
459     description: Query Project model

```

```
460     operationId: projectGet
461     tags:
462     - Project
463     security:
464     - BearerAuth: []
465     responses:
466     '200':
467         description: Success
468         content:
469             application/json:
470                 schema:
471                     type: array
472                     items:
473                         type: object
474                         $ref: 'models/Project.json#/response'
475         default:
476             description: Error
477             content:
478                 application/json:
479                     schema:
480                         $ref: '#/components/schemas/ErrorResponse'
481 /project/{id}:
482     patch:
483         description: Update a Project
484         operationId: projectPatch
485         tags:
486         - Project
487         security:
488         - BearerAuth: []
489         parameters:
490         - in: path
491           name: id
492           schema:
493               type: integer
494         requestBody:
495             content:
496                 application/json:
497                     schema:
498                         type: object
499                         $ref: 'models/Project.json#/request'
500         responses:
501         '200':
502             description: Success
```

```
503         content:
504             application/json:
505                 schema:
506                     $ref: 'models/Project.json#/response'
507     default:
508         description: Error
509         content:
510             application/json:
511                 schema:
512                     $ref: '#/components/schemas/ErrorResponse'
513
514 delete:
515     description: Destroy a Project
516     operationId: projectDelete
517     tags:
518         - Project
519     security:
520         - BearerAuth: []
521     parameters:
522         - in: path
523           name: id
524           schema:
525               type: integer
526     responses:
527         '200':
528             description: Success
529             content:
530                 application/json:
531                     schema:
532                         $ref: 'models/Project.json#/response'
533         default:
534             description: Error
535             content:
536                 application/json:
537                     schema:
538                         $ref: '#/components/schemas/ErrorResponse'
539 get:
540     description: Get given Project information
541     operationId: projectGetOne
542     tags:
543         - Project
544     security:
545         - BearerAuth: []
```

```
546     parameters:
547       - in: path
548         name: id
549         schema:
550           type: integer
551     responses:
552       '200':
553         description: Success
554         content:
555           application/json:
556             schema:
557               $ref: 'models/Project.json#/response'
558       default:
559         description: Error
560         content:
561           application/json:
562             schema:
563               $ref: '#/components/schemas/ErrorResponse'
564 /project/count:
565   get:
566     description: Get given Project information
567     operationId: projectGetOne
568     tags:
569       - Project
570     security:
571       - BearerAuth: []
572     responses:
573       '200':
574         description: Success
575         content:
576           application/json:
577             schema:
578               properties:
579                 count:
580                   description: Number of projects
581                   type: integer
582       default:
583         description: Error
584         content:
585           application/json:
586             schema:
587               $ref: '#/components/schemas/ErrorResponse'
588 /stats/project/{id}:
```

```
589   get:
590     description: Get statistics about fiberfy projects.
591     operationId: projectGetOne
592     tags:
593       - Statistics
594     security:
595       - BearerAuth: []
596     parameters:
597       - in: path
598         name: id
599         schema:
600           type: integer
601     responses:
602       '200':
603         description: Success
604         content:
605           application/json:
606             schema:
607               $ref: '#/components/schemas/StatsProjectResponse'
608       default:
609         description: Error
610         content:
611           application/json:
612             schema:
613               $ref: '#/components/schemas/ErrorResponse'
614 /maps/auth:
615   get:
616     description: Get all map sources and layers.
617     operationId: mapsGetListAuth
618     tags:
619       - Maps
620     security:
621       - BearerAuth: []
622     responses:
623       '200':
624         description: Success
625         content:
626           application/json:
627             schema:
628               $ref: '#/components/schemas/MapListResponse'
629       default:
630         description: Error
```



```
631         content:
632           application/json:
633             schema:
634               $ref: '#/components/schemas/ErrorResponse'
635 /maps/noauth:
636   get:
637     description: Get all map sources and layers with no
638       authentication needed.
639     operationId: mapsGetListNoAuth
640     tags:
641       - Maps
642     responses:
643       '200':
644         description: Success
645         content:
646           application/json:
647             schema:
648               $ref: '#/components/schemas/MapListResponse'
649         default:
650           description: Error
651           content:
652             application/json:
653               schema:
654                 $ref: '#/components/schemas/ErrorResponse'
655 /maps/wms/auth/{id}:
656   get:
657     description: Consume an internal WMS service with
658       authorization.
659     operationId: mapsGetWMS
660     tags:
661       - Maps
662     security:
663       - BearerAuth: []
664     parameters:
665       - in: path
666         name: id
667         description: Map id
668         schema:
669           type: integer
670       - in: query
671         name: token
672         description: Given JWT token to authorize operation
673         schema:
```

```
672         type: string
673     responses:
674         '200':
675             description: Success
676             content:
677                 image/png:
678                     schema:
679                         type: string
680                         format: binary
681         default:
682             description: Error
683             content:
684                 application/json:
685                     schema:
686                         $ref: '#/components/schemas/ErrorResponse'
687 /maps/wms/noauth/{id}:
688     get:
689         description: Consume an internal WMS service without
690             authorization.
691         operationId: mapsGetWMS
692         tags:
693             - Maps
694         parameters:
695             - in: path
696               name: id
697               description: Map id
698               schema:
699                   type: integer
700             - in: query
701               name: token
702               description: Given JWT token to authorize operation
703               schema:
704                   type: string
705         responses:
706             '200':
707                 description: Success
708                 content:
709                     image/png:
710                         schema:
711                             type: string
712                             format: binary
713         default:
714             description: Error
```

```
714         content:
715             application/json:
716                 schema:
717                     $ref: '#/components/schemas/ErrorResponse'
718 /swagger:
719     x-swagger-pipe: swagger_raw
720 components:
721     securitySchemes:
722         BearerAuth:
723             type: http
724             scheme: bearer
725     schemas:
726         LoginResponse:
727             required:
728                 - token
729             properties:
730                 token:
731                     type: string
732                 flag:
733                     type: boolean
734                 enum:
735                     - true
736         user:
737             type: object
738             $ref: 'models/User.json#/response'
739     ImportResponse:
740         properties:
741             sites:
742                 type: array
743                 items:
744                     type: object
745                     $ref: 'models/Site.json#/response'
746         paths:
747             type: array
748             items:
749                 type: object
750                 $ref: 'models/Path.json#/response'
751     ExportResponse:
752         type: object
753     StatsProjectResponse:
754         properties:
755             sites:
756                 type: array
```

```
757     items:
758         type: object
759     properties:
760         location_type:
761             type: string
762         count:
763             type: number
764 paths:
765     type: array
766     items:
767         type: object
768     properties:
769         type:
770             type: string
771         distance:
772             type: number
773 cables:
774     type: array
775     items:
776         type: object
777     properties:
778         type:
779             type: string
780         distance:
781             type: number
782 MapListResponse:
783     type: array
784     items:
785         type: object
786     properties:
787         layers:
788             type: array
789         items:
790             type: object
791         properties:
792             id:
793                 type: integer
794             name:
795                 type: string
796             description:
797                 type: string
798             isBase:
799                 type: boolean
```

```

800         options:
801             type: object
802     id:
803         type: integer
804     name:
805         type: string
806     description:
807         type: string
808     type:
809         type: string
810         enum: [tiles , wms]
811     internal:
812         type: boolean
813     options:
814         type: object
815     url:
816         type: string
817 ErrorResponse:
818     required:
819     - msg
820     properties:
821     msg:
822         type: string

```

Script extracció

```

1 // Script to extract project Models to json (swagger)
2
3 process.chdir(__dirname);
4
5 const fs = require('fs');
6
7 var models = require('require-all')({
8     dirname      : __dirname + '/api/models',
9     excludeDirs  : /^\. (git|svn)$/,
10    recursive     : true
11 });
12
13 function addAttributeToResult (name, attr, data, type) {
14     switch (attr.type) {
15         case 'string':
16         case 'number':
17         case 'integer':
18         case 'boolean':
19         case 'array':

```

```

20     case 'object':
21         data[type].properties[name] = {type: attr.type};
22         break;
23     case 'json':
24         data[type].properties[name] = {type: 'object'};
25         break;
26     default:
27         break;
28 }
29 }
30
31 function attachAdditionalInfoToAttribute (name, model, data, type) {
32     try {
33         let additional = model._swagger.attributes.append[type][name]
34         let current = data[type].properties[name];
35         current = Object.assign(current, additional);
36     } catch (err) {
37
38     }
39 }
40
41 function addAttributes (model, type, data) {
42     for (let y in modelsDefaultAttributes) {
43         if (modelsDefaultAttributes.hasOwnProperty(y)) {
44             let attr = modelsDefaultAttributes[y];
45             let isDisabled = false;
46             try {
47                 isDisabled = Boolean(model._swagger.attributes.ignore[type][y]);
48             } catch (err) {
49                 isDisabled = false
50             }
51             if (model.attributes[y] !== false && !isDisabled) {
52                 addAttributeToResult(y, attr, data, type);
53                 attachAdditionalInfoToAttribute(y, model, data, type);
54             }
55         }
56     }
57
58     for (let y in model.attributes) {
59         if (model.attributes.hasOwnProperty(y)) {
60             let attr = model.attributes[y];
61             let isDisabled = false;
62             try {
63                 isDisabled = Boolean(model._swagger.attributes.ignore[type][y]);
64             } catch (err) {
65                 isDisabled = false
66             }
67             if (!isDisabled) {
68                 addAttributeToResult(y, attr, data, type);
69                 attachAdditionalInfoToAttribute(y, model, data, type);
70             }
71         }

```

```

72 }
73 }
74
75 function addCustomAttributes (model, type, data) {
76   let attributes;
77   try {
78     attributes = model._swagger.attributes.additions[type];
79   } catch (err) {
80     // Nothing
81     return;
82   }
83   for (let y in attributes) {
84     if (attributes.hasOwnProperty(y)) {
85       let attr = attributes[y];
86       data[type].properties[y] = attr;
87     }
88   }
89 }
90
91 var modelsDefaultAttributes = require(__dirname + '/config/models').models.
  attributes;
92
93 for (let x in models) {
94   if (models.hasOwnProperty(x)) {
95     let data = {
96       request: {
97         properties: {}
98       },
99       response: {
100         properties: {}
101       }
102     };
103     let model = models[x];
104
105     addAttributes(model, 'request', data);
106     addAttributes(model, 'response', data);
107
108     addCustomAttributes(model, 'request', data);
109     addCustomAttributes(model, 'response', data);
110
111     let filename = __dirname + '/assets/swagger/models/' + x + '.json';
112     let json = JSON.stringify(data);
113     fs.writeFileSync(filename, json);
114   }
115 }

```

F.4 Importador

```

1 function importSites (data, project, zone, threshold) {
2   return new Promise((resolve, reject) => {

```

```

3  try {
4      if (data.type !== 'FeatureCollection') {
5          reject('GeoJSON_its_not_well_formatted')
6      }
7      let sites = []
8      let paths = []
9      let positions = data.features
10     for (let x in positions) {
11         // We iterate geojson
12         if (positions[x].type !== 'Feature') {
13             reject('GeoJSON_its_not_well_formatted')
14         }
15         if (positions[x].geometry.type === 'Point') {
16             // We import this site
17             let site = {}
18             site['type'] = 'notdefined' // Define default not hardcoded
19             if (positions[x].properties) { // We have to test if it's
20                 // possible somewhere
21                 if (positions[x].properties.name) site['name'] = positions[x].
22                     properties.name || 'site-' + uuidv1()
23                 if (positions[x].properties.type) site['type'] = positions[x].
24                     properties.type
25                 if (positions[x].properties.status) site['status'] = positions[
26                     x].properties.status
27             }
28             site['latitude'] = positions[x].geometry.coordinates[1]
29             site['longitude'] = positions[x].geometry.coordinates[0]
30             site['project'] = project
31             site['zone'] = zone
32             // Call database ORM
33             if (!isNaN(site.latitude) && !isNaN(site.longitude)) {
34                 sites.push(site)
35             }
36         }
37     }
38     else if (positions[x].geometry.type === 'LineString') {
39         // Test minimum 2
40         let path = {}
41         let flag = true
42         path['name'] = 'path-' + uuidv1()
43         path['type'] = 'notdefined'
44         path['status'] = 'Planned'
45         path['intermedial'] = []
46         for (let y in positions[x].geometry.coordinates) {
47             let latitude = positions[x].geometry.coordinates[y][1]
48             let longitude = positions[x].geometry.coordinates[y][0]
49             if (isNaN(latitude) || isNaN(longitude)) flag = false
50             path.intermedial.push([
51                 latitude,
52                 longitude
53             ])
54         }
55     }
56 }

```



```

51         if (flag) {
52             paths.push(path)
53         }
54     }
55 }
56
57 let sitesPromises = []
58 let pathsPromises = []
59 // Now we have to work with paths and sites
60 for (let x in paths) {
61     // Search first/end point existence
62     // If exists use existing point
63     let posIni = sites.findIndex(function (e, index) {
64         let dist = geolib.getDistance(
65             {latitude: paths[x].intermedial[0][0],
66              longitude: paths[x].intermedial[0][1]},
67             {latitude: e.latitude, longitude: e.longitude}
68         )
69         return (dist < threshold)
70     })
71     let posEnd = sites.findIndex(function (e, index) {
72         let end = paths[x].intermedial.length - 1
73         let dist = geolib.getDistance(
74             {latitude: paths[x].intermedial[end][0],
75              longitude: paths[x].intermedial[end][1]},
76             {latitude: e.latitude, longitude: e.longitude}
77         )
78         return (dist < threshold)
79     })
80     if(posIni < 0) { // In this case we have to create a Site
81         let newSite = {}
82         newSite['name'] = 'site-' + uuidv1()
83         newSite['type'] = 'notdefined'
84         newSite['latitude'] = paths[x].intermedial[0][0]
85         newSite['longitude'] = paths[x].intermedial[0][1]
86         newSite['project'] = project
87         newSite['status'] = 'Planned'
88         newSite['zone'] = zone
89         sites.push(newSite)
90         posIni = sites.length - 1
91     }
92     if(posEnd < 0) { // In this case we have to create a Site
93         let newSite = {}
94         let end = paths[x].intermedial.length - 1
95         newSite['name'] = 'site-' + uuidv1()
96         newSite['type'] = 'notdefined'
97         newSite['latitude'] = paths[x].intermedial[end][0]
98         newSite['longitude'] = paths[x].intermedial[end][1]
99         newSite['project'] = project
100        newSite['status'] = 'Planned'
101        newSite['zone'] = zone
102        sites.push(newSite)

```

```

103     posEnd = sites.length - 1
104   }
105   paths[x].posIni = posIni
106   paths[x].posEnd = posEnd
107 }
108 for (let x in sites) {
109   sitesPromises.push(Site.create(sites[x]))
110 }
111 let resultSites = []
112 Promise.all(sitesPromises).then(async function (values) {
113   for (let x in paths) {
114     let path = paths[x]
115     path.intermedial.splice(0, 1)
116     path.intermedial.splice(path.intermedial.length - 1, 1)
117
118     // We calculate distances
119     let distance
120     try {
121       distance = await sails.helpers.getDistancePath(values[path.
122         posIni].id, values[path.posEnd].id, path.intermedial)
123     } catch (err) {
124       reject(err)
125     }
126
127     pathsPromises.push(Path.create({
128       name: path.name,
129       type: path.type,
130       intermedial: path.intermedial,
131       first: values[path.posIni].id,
132       last: values[path.posEnd].id,
133       status: path.status,
134       distance: distance,
135       project: project
136     })))
137     resultSites = values
138   }
139   Promise.all(pathsPromises).then(function (values) {
140     let result = {
141       sites: resultSites,
142       paths: values
143     }
144     resolve(result)
145   }).catch(function (reason) {
146     reject(reason)
147   })
148   }).catch(function (reason) {
149     reject(reason)
150   })
151 }
152 catch (err) {
153   reject(err)
154 }

```

```
154     })
155 }
```

G Posada a punt

G.1 Systemd unit

```
1 [Unit]
2 Description=Fiberfy server service unit
3 After=network.target
4
5 [Service]
6 Environment="NODE_ENV=production"
7 Type=simple
8 User=fiberfy
9 WorkingDirectory=/home/fiberfy/fiberfy-server/
10 ExecStart=/usr/bin/node app.js
11 Restart=on-abort
12
13 [Install]
14 WantedBy=multi-user.target
```

Llista d'acrònims

| | | | |
|-------|---|------|-----------------------------------|
| Wi-Fi | Wireless Fidelity | OSI | Open System Interconnection |
| FTTH | Fiber to the Home | VPS | Virtual Private Server |
| GPON | Gigabit Passive Optical Network | NOC | Network Operations Center |
| IEEE | Institute of Electrical and Electronics Engineers | CAD | Computer-Aided Design |
| DSL | Digital Subscriber Line | DOM | Document Object Model |
| ISP | Internet Services Provider | ORM | Object-relational Mapping |
| WMS | Web Map Service | MVC | Model View Controller |
| GIS | Geographic Information System | LIR | Local Internet Registry |
| RIPE | Réseaux IP Européens | GLIR | Guifi.net Local Internet Registry |
| SaaS | Software as a Service | JSON | JavaScript Object Notation |
| CPD | Centre de processament de dades | CRUD | Create, Read, Update i Destroy |
| | | MMU | Memory Management Unit |

Bibliografia

- [1] atom: pàgina principal. <https://atom.io/>. Visitat: 2018-09-24.
- [2] Git project: pàgina principal. <https://git-scm.com/>. Visitat: 2018-09-23.
- [3] GitHub platform: pàgina principal. <https://github.com/>. Visitat: 2018-09-23.
- [4] Pàgina web producte FiberBase®. <http://fiberbasenms.com/why-fiberbase/>. Visitat: 2018-09-23.
- [5] Pàgina web producte VETROFiberMap™. <https://www.vetrofibermap.com/>. Visitat: 2018-09-22.
- [6] Que es un tablero kanban? <https://kanbantool.com/es/tablero-kanban>. Visitat: 2018-09-23.
- [7] What is scrum? <https://es.atlassian.com/agile/scrum>. Visitat: 2018-09-23.
- [8] Col·laboradors de la Wikipedia. Docker (software). [https://es.wikipedia.org/wiki/Docker_\(software\)](https://es.wikipedia.org/wiki/Docker_(software)). Visitat: 2018-09-23.
- [9] Col·laboradors de la Wikipedia. Estàndard IEEE 802.11. https://ca.wikipedia.org/wiki/IEEE_802.11. Visitat: 2018-09-22.
- [10] Col·laboradors de la Wikipedia. Fibra òptica. https://ca.wikipedia.org/wiki/Fibra_%C3%B2ptica. Visitat: 2018-09-22.
- [11] Col·laboradors de la Wikipedia. Mapatge d'objectes relacional. https://ca.wikipedia.org/wiki/Mapatge_d%27objectes_relacional. Visitat: 2018-12-24.
- [12] Col·laboradors de la Wikipedia. Modelo OSI. https://es.wikipedia.org/wiki/Modelo_OSI. Visitat: 2018-12-29.
- [13] Col·laboradors de la Wikipedia. Passive Optical Network. <https://ca.wikipedia.org/wiki/PON>. Visitat: 2018-09-22.
- [14] Col·laboradors de la Wikipedia. REST. <https://ca.wikipedia.org/wiki/REST>. Visitat: 2018-12-27.

- [15] Comunitat. Què és Guifi.net? https://guifi.net/ca/que_es. Visitat: 2018-09-22.
- [16] Docker Inc. Docker Registry. <https://docs.docker.com/registry/#what-it-is>. Visitat: 2018-12-02.
- [17] Red Hat Inc. Cap 13. protocolo ligero de acceso a directorios (ldap). In *Red Hat Enterprise Linux 4: Manual de referencia*, 2005.
- [18] Joan Manuel Tresserras i Gaju. RESOLUCIÓ CMC/3387/2007 Premis Nacionals Comunicació. http://dogc.gencat.cat/ca/pdogc_canals_interns/pdogc_resultats_fitxa/?action=fitxa&mode=single&documentId=445009&language=ca_ES, October 2007. Visitat: 2018-09-22.
- [19] MapServer contributors. MapServer front page. <https://mapserver.org/>. Visitat: 2018-11-24.
- [20] Roger Garcia Ferré. Entorns de desenvolupament en Docker. <http://ca.wiki.guifi.net/wiki/Fitxer:Docke-intro.pdf>. Visitat: 2018-11-24.
- [21] SmartBear Software. API Design Tools — Swagger. <https://swagger.io/solutions/api-design/>. Visitat: 2018-12-27.
- [22] SQLite contributors. SQLite front page. <https://www.sqlite.org/index.html>. Visitat: 2018-12-24.
- [23] SUSE. 25.2. structure of an ldap directory tree. In *SUSE LINUX Documentation (10.1)*.
- [24] Treballadors de la fundació Guifi.net. Història de Guifi.net. <https://fundacio.guifi.net/page/historia>. Visitat: 2018-09-22.
- [25] Vladimir Agafonkin. Leaflet.js front page. <https://leafletjs.com/>. Visitat: 2018-12-28.
- [26] Vue.js community. Vue.js — Introduction. <https://vuejs.org/v2/guide/index.html>. Visitat: 2018-12-28.